



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## Estudo de viabilidade tecnológica de uma análise semi-automática da colaboração no Google Docs

Luiz Eduardo Moreira

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientadora  
Prof.<sup>a</sup> Dr.<sup>a</sup> Germana Menezes da Nóbrega

Brasília  
2013

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Coordenador: Prof.<sup>a</sup> Dr.<sup>a</sup> Maristela Terto de Holanda

Banca examinadora composta por:

Prof.<sup>a</sup> Dr.<sup>a</sup> Germana Menezes da Nóbrega (Orientadora) — CIC/UnB

Prof.<sup>a</sup> Dr.<sup>a</sup> Maristela Terto de Holanda — CIC/UnB

Prof. Dr. Fernando William Cruz — FGA/UnB

### **CIP — Catalogação Internacional na Publicação**

Moreira, Luiz Eduardo.

Estudo de viabilidade tecnológica de uma análise semi-automática da colaboração no Google Docs / Luiz Eduardo Moreira. Brasília : UnB, 2013.

129 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2013.

1. Google Docs, 2. Gerência de Compromissos, 3. Escrita Colaborativa

CDU 004.4

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Estudo de viabilidade tecnológica de uma análise semi-automática da colaboração no Google Docs

Luiz Eduardo Moreira

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Prof.<sup>a</sup> Dr.<sup>a</sup> Germana Menezes da Nóbrega (Orientadora)  
CIC/UnB

Prof.<sup>a</sup> Dr.<sup>a</sup> Maristela Terto de Holanda    Prof. Dr. Fernando William Cruz  
CIC/UnB    FGA/UnB

Prof.<sup>a</sup> Dr.<sup>a</sup> Maristela Terto de Holanda  
Coordenador do Bacharelado em Ciência da Computação

Brasília, 04 de setembro de 2013

# Dedicatória

Dedico primeiramente ao meu pai, minha mãe e meu irmão, pois são a base para meu crescimento e aprendizagem. Dedico à Vanessa e sua família pelo apoio durante essa fase final do curso. E por fim, dedico aos meus colegas de curso que participaram dessa fase da minha vida.

# Agradecimentos

Agradeço à Prof.<sup>a</sup> Dr.<sup>a</sup> Germana pela dedicação e empenho durante esse trabalho e ao Prof. Dr. Fernando Cruz pelas dicas e conversas, que serviram de grande ajuda e estímulo.

# Abstract

Esta monografia está relacionada com dois projetos, o EngageGrid, e sua continuação EngageGrid+. O primeiro projeto tem como objetivo a criação de um sistema, conhecido como rede de compromissos, capaz de prover a gerência de compromissos numa comunidade de usuários. Esse sistema é baseado em dois elementos, os aspectos teóricos e a parte técnica, representada pela plataforma AGORA/GSD. O segundo projeto visa o desacoplamento da rede de compromissos da plataforma citada anteriormente e transporta-la para um ambiente em nuvem. Essa escolha permite que os serviços da rede de compromissos sejam acessados por qualquer plataforma colaborativa além do AGORA/GSD. Assim, a pesquisa desta monografia tem como objetivo analisar a API do Google Docs e verificar a viabilidade de uma interface entre seus serviços e os da rede de compromissos. Os principais pontos dessa análise de viabilidade são o compartilhamento e gerência de documentos de texto e a capacidade de captura de dados relacionada com o histórico de revisão. Um protótipo da rede de compromissos com interface aos serviços do Google Docs foi implementado para exemplificar a sua capacidade diante do contexto do projeto EngageGrid+.

**Palavras-chave:** Google Docs, Gerência de Compromissos, Escrita Colaborativa

# Abstract

This monograph is related to two projects, the EngageGrid and its continuation EngageGrid+. The first project aims to create a system, known as network of commitments, able to provide commitment management to a community of users. This system is based on two elements, the theoretical aspects and the technical part, represented by the platform AGORA/GSD. The second project aims at decoupling the network of commitments from aforementioned platform and transports it to a cloud environment. This choice allows the services of network of commitments to be accessed by any collaborative platform beyond the AGORA/GSD. Thus, the research of this thesis is to analyze the Google Docs API and verify the feasibility of an interface between its services and network of commitments. The main points of this analysis viability are the sharing and management of text documents and ability to capture data related to the revision history. A prototype of network of commitments with a interface to the services of Google Docs was implemented to exemplify its capacity before the project EngageGrid+ context.

**Keywords:** Google Docs, Commitment Management, Colaborative Writing

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Metodologia . . . . .	2
1.3	Estrutura do Trabalho . . . . .	2
<b>2</b>	<b>Revisão de Literatura: Abordagens Recentes para Análise da Escrita Colaborativa com Google Docs</b>	<b>3</b>
2.1	Glosser . . . . .	3
2.1.1	A Ferramenta . . . . .	4
2.1.2	Avaliação da Ferramenta . . . . .	6
2.1.3	Discussão sobre a Ferramenta . . . . .	6
2.2	WriteProc . . . . .	6
2.2.1	Ferramenta . . . . .	7
2.2.2	Discussão sobre a Ferramenta . . . . .	8
2.3	Estudo de Processos de Escrita Individual e Colaborativa . . . . .	8
2.3.1	Atividades da Escrita Colaborativa . . . . .	9
2.3.2	Estudo de Caso . . . . .	11
2.3.3	Discussão da Ferramenta . . . . .	14
2.4	Discussão Final do Capítulo . . . . .	14
<b>3</b>	<b>Referencial Teórico: Pilares para Análise Semi-automática da Colaboração no Google Docs</b>	<b>15</b>
3.1	Colaboração Apoiada por Computador . . . . .	15
3.1.1	CSCW . . . . .	15
3.1.2	eCSCW . . . . .	16
3.1.3	CRIWG . . . . .	16
3.1.4	CSCL . . . . .	16
3.1.5	SBSC . . . . .	17
3.2	Fomento à Colaboração em Nuvem no Projeto EngageGrid+ . . . . .	17
3.2.1	EngageGrid . . . . .	17
3.2.2	EngageGrid+ . . . . .	19
3.3	Google Docs: Capacidades e Limites . . . . .	20
3.3.1	Perspectiva do Usuário . . . . .	20
3.3.2	Perspectiva do Desenvolvedor . . . . .	23
3.4	Modelos para Análise da Colaboração . . . . .	32
3.4.1	Modelo de Análise do Sistema DEGREE . . . . .	32



3.4.2	Modelo de Análise 3C . . . . .	39
3.5	Considerações Finais sobre o Capítulo . . . . .	41
<b>4</b>	<b>Concepção e Prototipagem de Interface entre Serviços da Rede de Com-</b>	
	<b>promissos e Google Docs: Uma Primeira Versão</b>	<b>42</b>
4.1	Arquitetura . . . . .	43
4.1.1	Organização Física . . . . .	43
4.1.2	Organização Conceitual . . . . .	45
4.2	Protótipo . . . . .	45
4.2.1	Modelos de Análise e o Cálculo da Colaboração . . . . .	47
<b>5</b>	<b>Conclusão</b>	<b>53</b>
5.1	Conclusão . . . . .	53
5.2	Trabalhos Futuros . . . . .	54
	<b>Referências</b>	<b>55</b>

# Lista de Figuras

2.1	Interface gráfica do sistema Glosser. Fonte: [24]. . . . .	5
2.2	Arquitetura do sistema WriteProc. Fonte:[21]. . . . .	8
2.3	Gráfico pontilhado da ferramenta ProM. Fonte: [22]. . . . .	12
2.4	Padrões de caminhos. Fonte: [22]. . . . .	13
3.1	Etapas do ciclo de vida e sua relação com a base de dados. Fonte: [14]. . .	19
3.2	A tela da interface de gerência. (A) Lista de recursos. (B) <i>Labels</i> , (C) Botões de criar e <i>upload</i> de arquivos, (D) Botões de configuração (E) Barra de busca. . . . .	22
3.3	A tela da interface de Edição. (A) A área de edição, (B) a barra de ferramentas, (C) os menus e (D) a barra de revisão de histórico. . . . .	23
3.4	Diagrama de sequência para autorização usando OAuth 2.0. Fonte: <a href="https://developers.google.com/accounts/docs/OAuth2?hl=pt-BR">https://developers.google.com/accounts/docs/OAuth2?hl=pt-BR</a> . . . . .	25
3.5	Definição do conceito de atividade. Fonte: [10]. . . . .	33
3.6	Um exemplo de conversa semi-estruturada. Fonte: [10]. . . . .	35
3.7	Atributos de análise de um grupo. Fonte: [10]. . . . .	37
3.8	Atributos de análise de um indivíduo. Fonte: [10]. . . . .	38
4.1	Interação entre a rede de compromissos e a API do Google Docs. . . . .	42
4.2	Organização dos recursos do protótipo. . . . .	44
4.3	Arquitetura conceitual da aplicação que é representada pelos três módulos: “servlet”, “model” e “gDocs”. . . . .	46
4.4	Tela de <i>login</i> do protótipo. . . . .	47
4.5	Tela de <i>login</i> na API do Google Docs. . . . .	47
4.6	Tela de gerência do protótipo. . . . .	48
4.7	Tela de configuração do protótipo. . . . .	48
4.8	Tela de negociação do protótipo. . . . .	49
4.9	Tela de realização do protótipo. . . . .	49
4.10	Tela de análise do protótipo. . . . .	50
4.11	Reutilização do modelo DEGREE para este trabalho. . . . .	52

# Lista de Tabelas

- 2.1 Tabela da heurística contendo as atividades da escrita colaborativa. Fonte: [22]. 10

# Capítulo 1

## Introdução

As ferramentas de edição de texto sofreram uma grande evolução durante esses últimos anos. As suas primeiras edições eram executadas em terminais por um único indivíduo. A evolução dessas ferramentas trouxe melhorias na interface gráfica e na capacidade de manipulação do texto. Um dos últimos avanços relacionado a elas foi a disponibilização de seus serviços na *web*.

O Google Docs[5] é o melhor exemplo de ferramenta de edição na *web*. Seu serviço não se restringe somente em disponibilizar o ambiente de edição. Também há o compartilhamento, armazenamento e a edição colaborativa. Diante desses serviços, o Google Docs se mostra uma ferramenta poderosa para colaboração, porém não há um suporte para a gestão de compromissos firmados nessa plataforma.

Um ambiente de gestão de compromissos tem o objetivo de gerir os recursos relacionados a um compromisso e fornecer um *feedback* do processo para os colaboradores. Isso formaliza o processo de produção de trabalhos em qualquer ferramenta. A associação da gestão de compromissos com o Google Docs apresenta as seguintes vantagens:

- Formalização da produção de algum documento na plataforma;
- Gestão de pessoas e recursos;
- Estabelecimento formal de metas e objetivos;
- Análise do trabalho colaborativo e seu *feedback*;
- Registro do trabalho para fins de gestão de pessoas.

### 1.1 Objetivos

Assumindo que o Google Docs possui uma API[6] que permite acesso ao histórico de revisões, o objetivo geral desse trabalho é verificar a viabilidade da interface entre o Google Docs e o sistema de gestão de compromissos desenvolvido no projeto EngageGrid+ [12], conhecido como rede de compromissos.

Para atingir o objetivo geral, foram definidos alguns objetivos específicos:

1. Desenvolver um protótipo que demonstre essa interface;

2. Prover análise da colaboração de trabalhos desenvolvidos na plataforma do Google Docs (referente a etapa de análise do ciclo de vida de um compromisso da rede de compromissos).

A seguir, é classificado o trabalho desenvolvido e são descritos os capítulos seguintes.

## 1.2 Metodologia

A natureza deste trabalho pode ser classificada como aplicada. A API do Google Docs, os aspectos teóricos da rede de compromissos e alguns modelos de análise da colaboração são estudados com a finalidade de contribuir para o estudo de viabilidade das interfaces propostas nos objetivos. A bibliografia foi utilizada para pesquisar modelos para análise da colaboração de documentos escritos e também para achar evidências sobre as capacidades do Google Docs.

## 1.3 Estrutura do Trabalho

Esta monografia é composta de por cinco capítulos, incluindo este. No capítulo 2, é feita uma descrição de trabalhos que utilizam o Google Docs como ferramenta de edição e sobre a análise de processos da escrita colaborativa. No capítulo 3, são feitas a descrição das principais conferências sobre suporte ao trabalho colaborativo, a descrição dos projetos EngageGrid e EngageGrid+, a descrição da API do Google Docs e a descrição de dois modelos de análise da colaboração. O capítulo 4 trata dos aspectos técnicos e conceituais acerca do protótipo desenvolvido. Por fim, no capítulo 5, encerra-se o trabalho, apresentando conclusões, dificuldades e projetos futuros.

## Capítulo 2

# Revisão de Literatura: Abordagens Recentes para Análise da Escrita Colaborativa com Google Docs

Os avanços tecnológicos alteraram a forma como o homem vive em muitas áreas do seu cotidiano. Uma delas que mudou significativamente é a produção escrita. Essa mudança foi gradual, desde o uso de papel e lápis, seguiu-se com a introdução do teclado nas máquinas de escrever e foi otimizado com o surgimento dos computadores. Essa transição não foi meramente física, do papel para a tela, ela mudou também o modo como o ser humano interpreta e produz o conhecimento na forma escrita. Além de facilitar a inserção, deleção e edição de caracteres, permitiu outros modos de produção como o trabalho escrito em grupo de modo síncrono.

Esse capítulo faz uma revisão de trabalhos que estão relacionados com a ferramenta de edição do Google Docs. Ele está dividido em 3 seções, começando pela descrição do Glosser [24], uma ferramenta na área de suporte à escrita. A descrição do WriteProc [21], uma ferramenta de suporte à escrita individual e colaborativa que foca em análise dos processos de escrita. E por fim, a descrição de um trabalho que utiliza heurística para encontrar atividades de escrita através da captura de dados automaticamente. Esses trabalhos foram escolhidos, por explorarem a API e o caráter colaborativo do Google Docs.

### 2.1 Glosser

Em Glosser [24], o autor mostra como a escrita para o aprendizado é comum entre várias disciplinas no ensino superior e que as habilidades adquiridas através da escrita são facilmente aplicadas em outras áreas do conhecimento. Portanto, a escrita é uma parte importante para o currículo dos estudantes na atualidade.

Algumas capacidades são esperadas dos aprendizes ao trabalharem na produção de um texto, como habilidades com o vocabulário e a gramática, além de outras que envolvem interpretação, como análise, argumentação e opinião em relação ao assunto tratado. Durante a escrita, é necessário que o estudante se envolva num pensamento reflexivo do trabalho que está sendo feito e do processo de escrita que deve ser seguido. Essa atividade é complexa e sem um suporte específico, pode gerar uma dificuldade na sua execução.

Algumas ferramentas de suporte surgiram para dar apoio à escrita em diferentes etapas do processo de escrita. Algumas delas fornecem *feedback* para o escritor, outras fazem avaliação automática do texto. Outras, conhecidas como *Automatic Essay Scoring*, focam na avaliação do documento. O autor argumenta que essas ferramentas levam em conta principalmente a questão com relação ao tempo, custo e confiabilidade. Portanto, elas deixam de focar no processo de escrita e focam somente no produto. Muitos desses sistemas oferecem pouco ou nenhum *feedback* para o estudante.

O autor mostra que o processo de escrita é formado por um conjunto de 3 sub-processos: pré-escrita, elaboração e revisão. Eles não seguem uma ordem sequencial e ocorrem unicamente para cada indivíduo engajado na escrita. Durante esse processo, o fornecimento de *feedback* é essencial para o processo de aprendizado. Ele também cita a ideia de incitar o processo reflexivo e para que isso ocorra de forma significativa, é necessário fazer o estudante pensar explicitamente.

### 2.1.1 A Ferramenta

O Glosser [24] é uma ferramenta projetada para dar suporte no ensinamento e aprendizado da escrita acadêmica em inglês. O sistema provê questionamentos chaves, que podem ser customizados para objetivos específicos, e *feedback* (que nessa ferramenta é chamado de “gloss”), que pode ser usado pelo estudante para refletir de forma mais eficiente e analisar o trabalho e o processo de escrita. O resultado esperado é um texto com maior qualidade e um fortalecimento no aprendizado.

Essa ferramenta utiliza técnicas de mineração de dados para obter informações semânticas sobre o texto. Basicamente, são utilizados dois métodos, a LSA (*Latent Semantic Analysis*) [17] e o algoritmo Lingo [20]. O primeiro tem objetivo de extrair sentenças chaves do texto, e também pode ser usado para calcular a distância semântica entre dois parágrafos. O segundo é utilizado para criar agrupamentos onde sentenças do texto se juntarão por tópicos. Como base de processamento textual, foi utilizado o software de indexação Lucene da Apache, onde são realizadas as operações *tokenizing*<sup>1</sup>, *stemming*<sup>2</sup> e remoção de palavras de parada.

O sistema tem dois objetivos principais. O primeiro é ativar uma reflexão sobre a qualidade da escrita. O segundo é dar suporte à reflexão no processo de escrita do aprendiz. O primeiro objetivo é pretendido através do uso de uma estrutura de critérios avaliativos chamada MASUS [11], criado como um pré-teste para medir habilidade na escrita acadêmica. Essa estrutura apresenta 5 atributos de qualidade:

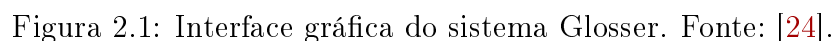
- Uso de fontes de pesquisa;
- Estruturação e desenvolvimento da resposta;
- Controle do estilo de escrita;
- Corretude gramatical;
- Qualidade da apresentação.

---

<sup>1</sup>Operação de separação de elementos com algum significado (cada elemento é chamado em inglês de *token*).

<sup>2</sup>Processo de reduzir cada palavra para seu radical.

A interface do sistema apresenta abas, cada uma referente a um dos 3 atributos citados anteriormente. Cada aba apresenta um conjunto de perguntas, que servirão para ativar a reflexão, e um conteúdo de suporte. Esse conteúdo pode ser textos ou imagens e provê ajuda para responder as perguntas referentes a sua respectiva aba. A figura 2.1 mostra uma captura de tela do sistema.



**Aba “*Structure*”:** Nessa aba, são exibidas perguntas que farão o estudante de fato analisar a estrutura do texto e checar se estão corretas. Por exemplo, uma das perguntas de suporte é “A conclusão decorre dos argumentos?”. O conteúdo de suporte são um campo, abaixo das perguntas e à esquerda da tela, contendo as sentenças chaves do texto e outro campo, a direita do campo citado anteriormente, onde se tem a conclusão apresentada no texto. Eles servem para o estudante avaliar se ele escreveu as evidências certas para sua conclusão.



**Aba “*Coherence*”:** Essa aba apresenta um conjunto de perguntas para reflexão, por exemplo, “Você entendeu como cada parágrafo e sentença segue do anterior?”.

O conteúdo de suporte nessa aba apresenta a identificação de pares consecutivos de parágrafos que estão distantes semanticamente de acordo com o LSA. Isso pode indicar incoerência ou falta de fluxo no texto.

**Aba “*Topics*”:** Uma das perguntas apresentadas nessa aba é “Essa dissertação apresenta trabalhos acadêmicos como fatos, ou analisa a sua importância e discute sua utilidade?”.

O conteúdo de suporte mostra um conjunto de tópicos que agrupa os conceitos identificados como os mais enfatizados para a versão do documento selecionado.

### 2.1.2 Avaliação da Ferramenta

Essa ferramenta foi aplicada num grupo de textos produzidos colaborativamente num Wiki por alunos matriculados numa disciplina da universidade de Sydney [24]. Eles serviram de base para fazer uma primeira avaliação da ferramenta. Dois aspectos são avaliados nessa análise inicial: a acurácia dos algoritmos e o impacto no aprendizado dos alunos.

Foram analisados os trechos retirados dos textos que servem de suporte para responder cada pergunta feita nas categorias citadas anteriormente. Inicialmente, foi preciso ajustar alguns parâmetros dos algoritmos e algumas questões foram rescritas. A ferramenta apresentou boa capacidade de selecionar sentenças apropriadas do texto. E o uso da aba “*Topics*” proveu uma boa lista de tópicos endereçando algumas sentenças em mais de um tópico, construindo grupos com mais significados.

### 2.1.3 Discussão sobre a Ferramenta

Essa ferramenta é interessante, pois muda a forma como era tratada a utilização da escrita para aprendizado, diferente do enfoque de muitas instituições que usam sistemas automatizados para correção de textos. O *feedback* fornecido de forma rápida e durante o processo de produção do texto, permite uma reflexão enquanto as ideias ainda estão em construção, estimulando um pensamento mais crítico e uma forma mais dinâmica na procura de novas ideias para se colocar no texto.

A abordagem com uso de mineração de texto também é outro ponto forte. Apesar da análise da proximidade semântica de trechos do texto dos aprendizes não estar de acordo com o esperado, essa fornece indícios. Por se tratar de uma primeira abordagem, a ferramenta vai passar por melhorias que futuramente podem apresentar resultados mais precisos diante do seu objetivo.

## 2.2 WriteProc

Em WriteProc [21], o autor aborda a escrita colaborativa citando o seu crescimento devido o avanço da internet, possibilitando inúmeras maneiras de compartilhar um arquivo. Destaca que é importante ter habilidades com a escrita individual e colaborativa nos cenários acadêmico, industrial e governamental.

Apesar dessa vasta extensão de áreas, o autor foca na educação. Ele cita o trabalho de Goldberg et al [16], onde foi coletada uma década de dados empíricos sobre suporte à escrita nessa área. Esse estudo percebeu que estudantes fazem mais revisões do texto durante a produção do que no fim quando estão produzindo por computador. Também notou-se que são feitas mais revisões durante as produções dos rascunhos quando produzidos por computador. E cita a importância da revisão de companheiros durante a escrita utilizada para aprendizagem.

O autor cita que existem muitos trabalhos em mineração de texto para melhorar o suporte à qualidade da escrita nas duas últimas décadas. Entretanto, ele destaca que trabalhos, como ferramentas de avaliação automática de texto, visualização e *clustering* de documentos focam somente no produto e não no processo em si. Portanto, busca-se investigar como ideias e conceitos que são desenvolvidos durante o processo de escrita podem ser usados para melhorar a qualidade do documento e as habilidades de escrita dos envolvidos.

### 2.2.1 Ferramenta

Em WriteProc [21], são desenvolvidos um *framework* conceitual e ferramentas para suporte da escrita colaborativa. Ele é baseado na taxonomia de escrita colaborativa proposta por Lowry [19] e define atividades de escrita, estratégias, modos de trabalho e papéis envolvidos na escrita colaborativa. Ele é constituído de três ferramentas, uma de escrita, uma de revisão e uma de análise.

A ferramenta utiliza o Google Docs como ambiente de desenvolvimento dos trabalhos escritos. É possível recuperar todas as revisões, podendo ser feito um histórico das ações realizadas em toda a colaboração. Os seguintes dados são fornecidos por unidade de revisão: o id do usuário, uma etiqueta contendo o tempo em que foi salva tal revisão e o número da revisão.

O Glosser é a ferramenta de *feedback* desse sistema e fornece assistência aos escritores na revisão de seus trabalhos e nas de colegas. As ações de revisão feitas nessa ferramenta são salvas num registro para posterior análise das mudanças alcançadas nos textos antes e depois de receberem *feedback*.

Para fazer a análise de processos, é utilizada uma ferramenta de mineração de processos chamada ProM [23]. Os dados usados pela ferramenta são retirados das revisões do Google Docs e da ferramenta Glosser.

Essa ferramenta é usada para descobrir as atividades de escrita e as interações entre os escritores. A ferramenta de análise de texto é utilizada para analisar as mudanças em cada revisão e descobrir a natureza e o nível das alterações. As informações dos textos, como a versão, são indexadas usando o Lucene da Apache e a biblioteca de mineração de dados utilizada é a TML [9] (*Text Mining Library*).

O sistema funciona com o Google Docs como meio para a escrita colaborativa. Os documentos produzidos são recuperados através da API do Google Docs e usados para revisão na ferramenta Glosser. Para realizar a análise proposta pelo WriteProc, são recolhidos os documentos e os *logs* do Google Docs e os *logs* do Glosser, como pode ser visto na figura 2.2.

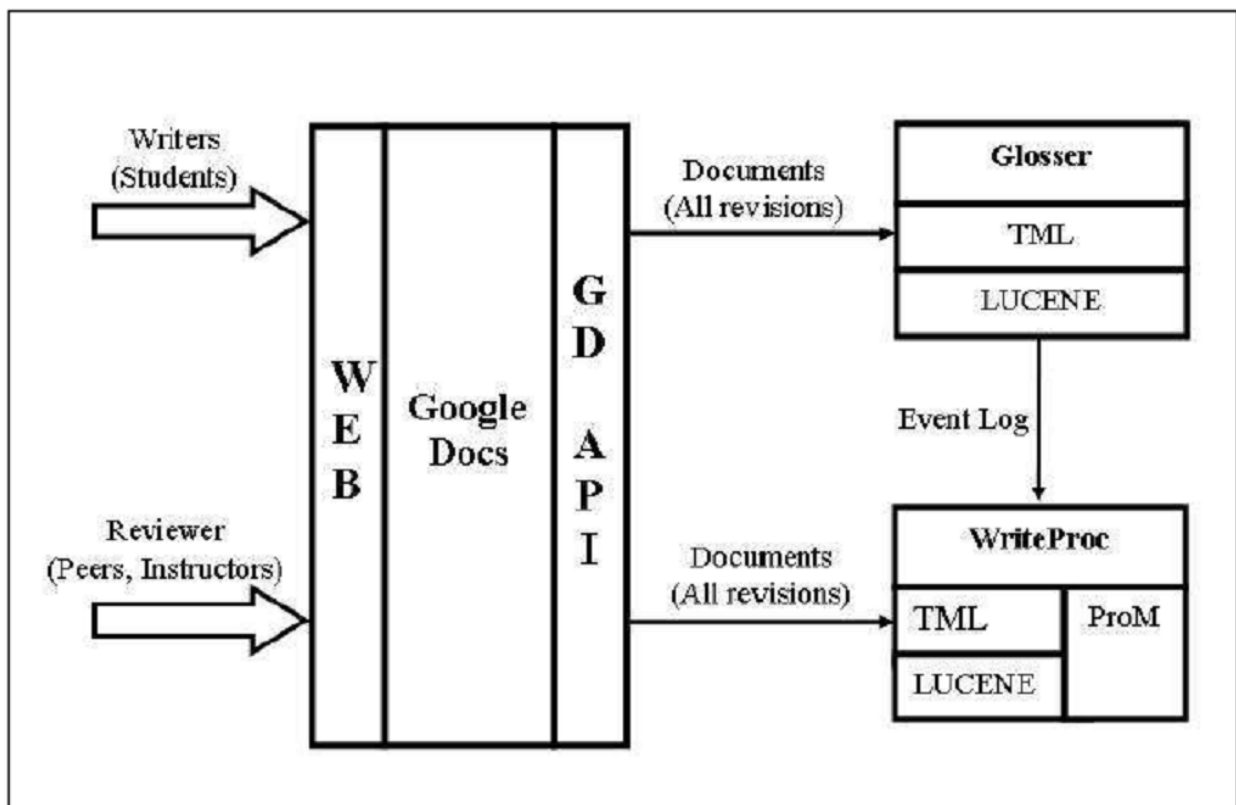


Figura 2.2: Arquitetura do sistema WriteProc. Fonte:[21].

### 2.2.2 Discussão sobre a Ferramenta

Essa ferramenta foi utilizada num estudo com alunos de um curso de graduação. O estudo não chegou a um resultado conclusivo com relação ao processo da escrita. Entretanto, foi satisfatória a coleta dos dados, mostrando os acessos à edição dos textos e o ciclo de acessos ao Glosser.

Apesar de não tirar resultados conclusivos, esse trabalho é um dos poucos a tratar da escrita colaborativa com enfoque na análise a partir de mineração de processos e dados. Outro ponto positivo é abordar o uso de um software colaborativo conhecido, o Google Docs.

## 2.3 Estudo de Processos de Escrita Individual e Colaborativa

A seção anterior descreveu as ferramentas utilizadas em alguns estudos que buscavam conhecer mais sobre os processos de escrita individual e colaborativa. Os resultados não mostraram com exatidão a existência de um padrão de processo na escrita. Contudo, em [22], é feita uma abordagem com a definição explícita das etapas no processo de escrita. Buscou-se com heurísticas definir as ações das pessoas durante a escrita. Além disso, as duas ferramentas previamente descritas nesse capítulo são utilizadas para alcançar tal objetivo.

Nesse estudo, o WriteProc [21] é utilizado para recuperar as revisões do Google Docs. Elas são salvas a cada pequena porção de tempo ou por comando do escritor e é produzida uma revisão a cada comando de salvamento. Para um documento em particular, é obtido o histórico de revisão, nele há informações como: id do usuário, id da sessão, data do registro (em milissegundos) e o número da revisão.

O histórico de revisão é usado para recuperar sessões de escrita individuais para cada escritor. A cada minuto de escrita, é feito o download da revisão de um escritor e seus dados são extraídos. Os parágrafos e sentenças retirados da revisão são indexados para análise. O WriteProc realiza uma análise baseada em texto nas revisões indexadas, onde são extraídos significados semânticos das mudanças textuais e identifica as atividades de escrita. Para realizar isso, são utilizadas heurísticas. E por fim, as atividades de escrita são analisadas utilizando-se mineração de processos.

### 2.3.1 Atividades da Escrita Colaborativa

Em *Building a taxonomy and nomenclature of collaborative writing to improve interdisciplinary research and practice* [19], foram definidas 6 atividades comuns na escrita colaborativa: *Brainstorming*, delineação, elaboração, reavaliação, revisão e edição. É importante notar que essas atividades não ocorrem numa sequência linear. Além disso, a atividade de reavaliação pode ser executada não só pelo escritor, mas por outras pessoas que não editam o texto. Abaixo segue a definição de cada uma das 6 atividades:

- **Brainstorming:** Atividade de desenvolver ideias e coloca-las num rascunho.
- **Delineamento:** Direcionamento do documento de uma forma menos detalhada, definindo algumas seções e subseções.
- **Elaboração:** A escrita da primeira versão formal do documento.
- **Reavaliação:** Etapa que há participação de um colega para ler e anotar pontos do texto que podem ser melhorados.
- **Revisão:** Baseado nos pontos abordados do item anterior, são feitas alterações no texto.
- **Edição:** Alterações finais no documento para torna-lo mais consistente.

Nesse trabalho, o foco foi na identificação automática das 5 seguintes atividades: *Brainstorming*, delineação, elaboração, revisão e edição.

Para identificá-las, foram usadas heurísticas básicas. Elas são baseadas em mudanças textuais, estruturas do texto, mudanças nos tópicos abordados e a melhoria da coesão em relação a duas revisões consecutivas. A tabela 2.1 esquematiza as categorias usadas nas heurísticas definidas nesse trabalho.

Nessa parte da seção, são descritas como as heurísticas são usadas para identificação das atividades.

#### Estruturas Textuais

As estruturas textuais podem ser definidas analisando a estrutura do texto escrito (S1) e o número de sentenças e parágrafos. Durante o *Brainstorming*, o escritor normalmente

Tabela 2.1: Tabela da heurística contendo as atividades da escrita colaborativa.  
Fonte: [22].

Writing activities	Surface change	Reorganization of information	Consolidation of information	Distribution of information	Addition of information	Deletion of information	Alteration of form (Macro-structure change)	Micro-structure change	Structure	Number of Sentence vs Number of Paragraphs	Ratio of Number of Words	Topic Overlap	Cohesion Measure
	C1	C2	C3	C4	C5	C6	C7	C8	S1	S2	F1	F2	F3
<b>B</b>	Y	Y	Y	Y	Y	Y	Y	Y	L	$S \approx P$	F	N/A	N/A
<b>O</b>	Y	Y	Y	Y	Y	Y	Y	Y	SL	$S \approx P$	F	N/A	N/A
<b>D</b>	N	N	N	N	Y	Y	N**	Y	Sec. & Par.	$S > P$	F	F	F
<b>R</b>	N	Y	Y	Y	N*	N*	Y	N*	Sec. & Par.	$S > P$	F	F	I/C
<b>E</b>	Y	N	N	N	N	N	N	N	Sec. & Par.	$S > P$	C	C	C

faz uma lista de palavras soltas ou pequenos grupos de palavras. Por causa disso, o número de sentenças é quase o mesmo do número de parágrafos. Já na fase de delineação, o número de sentenças e parágrafos permanece o mesmo, porém a estrutura do texto começa a sofrer alterações e há a organização do texto por seção e subseção. Quando o escritor começa a fase de elaboração, o número de sentenças e parágrafos mudam. É esperado que o número de sentenças seja bem maior que o número de parágrafos. Isso também se aplica para as fases de edição e revisão.

### Operações de Mudanças no Texto

As operações de mudanças de texto apresentadas na tabela 2.1 são: alteração da aparência(C1), reorganização da informação(C2), consolidação da informação(C3), distribuição da informação(C4), adição da informação(C5), deleção da informação(C6), macro mudanças na estrutura (C7) e micro mudanças na estrutura(C8). Algumas suposições foram feitas:

- Durante o *Brainstorming*, escritores podem reordenar, adicionar ou deletar itens da lista de ideias. Isso ocorre similarmente com a delineação.
- Durante a elaboração, revisão e edição, operações de mudança de texto se tornam mais complicadas. A atividade muda de algum estado (*Brainstorming* ou delineação) para elaboração quando a estrutura do texto escrito muda de lista de ideias (*Brainstorming*) ou de listas estruturadas (delineação) para parágrafos. Durante a fase de elaboração, informação é adicionada e removida o tempo todo, portanto, operações de mudança como expansão de informação, deleção de informação e pequenas alterações na estrutura implicam na atividade de elaboração.

- As 3 últimas operações de mudança citadas no item anterior podem significar outra atividade se elas ocorrerem depois da revisão feita por um companheiro. É assumido que depois de receber *feedback*, o escritor tende a adicionar, deletar ou alterar texto existente no documento. Pode acontecer também a completa reformulação do texto, representada pela operação macro alteração na estrutura.
- Atividades comuns de revisão são reordenar, consolidar e distribuir textos. Essas mudanças acontecem com frequência depois da atividade de elaboração e ocorrem muitas outras vezes no processo de escrita. Uma suposição nesse caso é que os escritores param de escrever e revisam o documento com frequência para haver uma melhora na coesão e uma melhor transmissão de ideias para o leitor.
- Todas as operações de mudanças de aparência, incluindo formatação, correção de escrita e pontuação são consideradas atividades de edição. As operações de C2 até C4 (tabela 2.1) são também consideradas atividades de edição e ocorrem várias vezes no processo de escrita.

## Número de Palavras

A relação entre o número de palavras entre duas revisões consecutivas é computado. Essa relação é utilizada em conjunto com a sobreposição de tópicos e a medida de coesão para determinar as atividades de escrita.

## Sobreposição de Tópicos

A mudança na abordagem de tópicos em duas revisões consecutivas é analisada. O autor usa essa medida, pois acha que quando um escritor aborda um tópico, ele continua escrever sobre esse assunto durante algumas revisões para manter a atenção do leitor. Ao identificar os conceitos abordados numa revisão, há a comparação entre duas revisões consecutivas para observar como o autor desenvolveu suas ideias e conceitos durante o processo de escrita. Durante a fase de elaboração e revisão, a sobreposição de tópicos muda drasticamente. Entretanto, durante a edição ela permanece constante.

## Mudanças na Coesão

A coesão é outra medida utilizada para detectar as atividades de escrita. Ela é baseada no nível de coesão entre duas revisões consecutivas. É calculada a distância entre sentenças e parágrafos consecutivos para ter uma noção de como os parágrafos e todo o texto foi escrito. Uma suposição é que durante a fase de elaboração, a coesão possua uma alta taxa de variação. Depois de uma atividade de revisão, a coesão aumenta muito e durante fases de edição não há variação significativa.

### 2.3.2 Estudo de Caso

Foi realizado um estudo com alunos de um curso da Universidade de Sydney, onde foram analisados textos exigidos numa disciplina chamada Análise e projeto de *E-Business*. Nessa disciplina, os alunos foram organizados em grupos e foi pedido para produzirem um documento de especificação de projeto contendo entre 1500 e 2000 linhas. É usada

também a revisão de colegas, onde dois integrantes de outro grupo leem o documento e dão o *feedback*. Depois da revisão feita por colegas, o grupo pode efetuar uma revisão do documento e as devidas alterações antes da submissão final. As notas juntamente com a qualidade de cada grupo através do semestre foram utilizadas para encontrar padrões de comportamento em relação a qualidade dos trabalhos apresentados.

## Análise do Processo de Escrita

Os dados coletados dos trabalhos relativos a 21 grupos retornaram 8233 eventos. Foram registrados 6 tipos diferentes de eventos representando os 6 tipos de atividades. O número médio de eventos foi de 392, apresentando o mínimo de 77 eventos e o máximo de 705 eventos. Nesse trabalho, são referenciados somente alguns dados da pesquisa para efeito de exemplificação da análise feita pelo autor.

Para efetuar a análise, foi utilizado o gráfico pontilhado da ferramenta ProM de análise de processos. Esse gráfico apresenta as atividades dos 21 grupos distribuídos durante o tempo de execução da atividade, como podem ser vistos na figura 2.3. As 6 atividades mencionadas no começo dessa seção estão presentes na imagem sendo representadas por formas e cores diferentes. Os círculos brancos representam o *Brainstorming*, o círculos marrons representam delineação, os triângulos pretos representam elaboração, os quadrados pretos representam reavaliação, os quadrados marrons representam revisão e os diamantes brancos denotam atividades de edição.

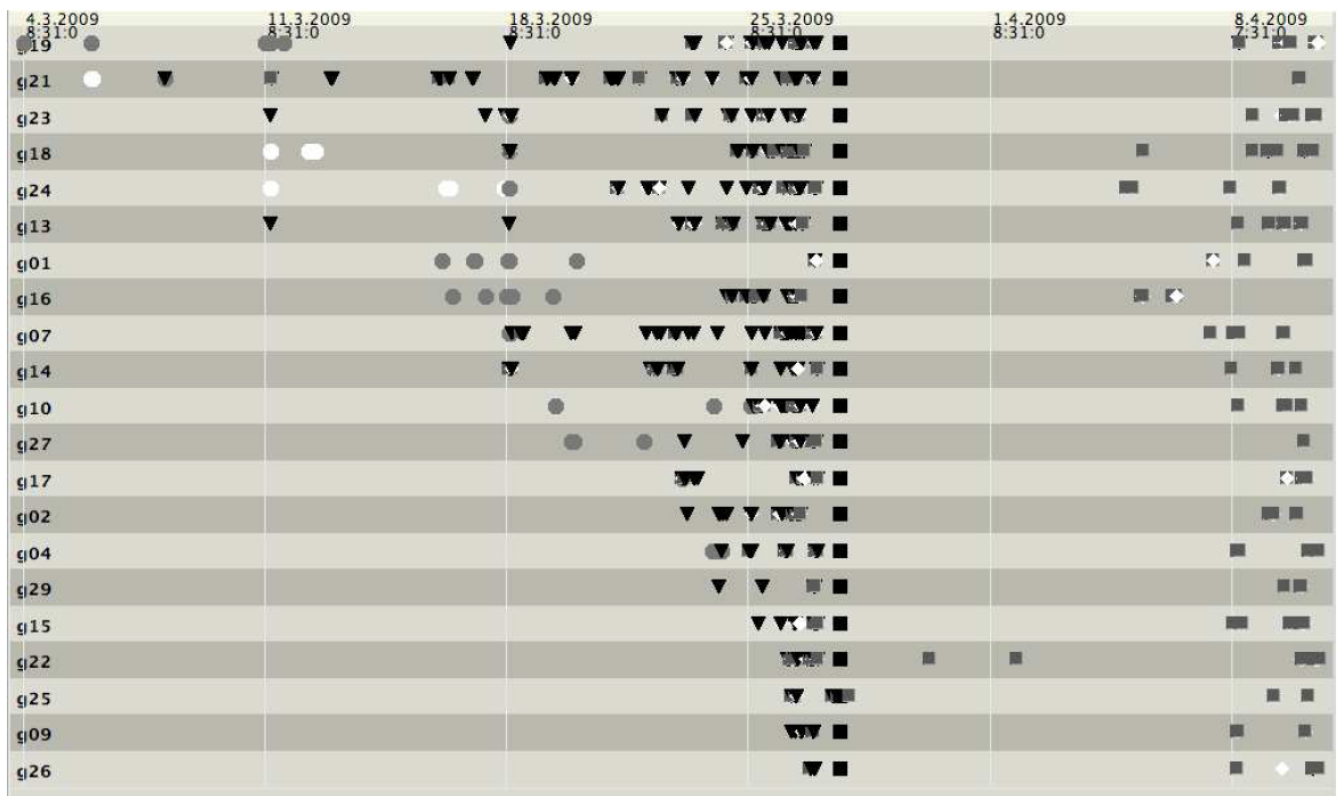


Figura 2.3: Gráfico pontilhado da ferramenta ProM. Fonte: [22].

Por meio do gráfico, foi possível analisar o comportamento dos grupos, notando a data de início dos trabalhos, número de atividades e quais delas foram executadas. A verificação busca encontrar uma relação entre as notas, principalmente as baixas e as altas, e algum padrão nas atividades desenvolvidas.

Os grupos que fizeram *Brainstorming* e delineação começaram suas atividades bem mais cedo e tiveram notas melhores. Entretanto, alguns grupos começaram suas atividades poucas semanas antes da submissão final e apresentaram notas altas. Isso ocorreu, pois esses grupos realizaram as atividades *Brainstorming* e delineação fora da ferramenta. Os alunos desses grupos começaram as suas produções utilizando um editor de texto como o MS Word, pois faltando uma semana para a submissão de revisão por outros grupos (no meio da semana de 25/03 até 01/04) foram colocados grandes volumes de texto.

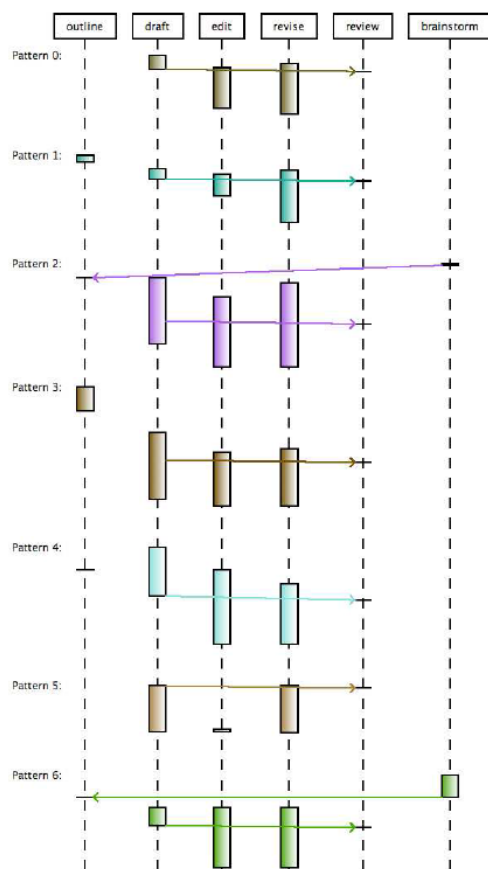


Figura 2.4: Padrões de caminhos. Fonte: [22].

Outro item importante da pesquisa foi o uso de um *plugin* da ferramenta ProM para traçar caminhos mais frequentes das atividades de escrita, veja a figura 2.4. Foram descobertos 7 caminhos diferentes. Deles, 5 desses caminhos são de grupos que não fizeram o *Brainstorming* e a delineação antes da elaboração. Mais da metade da classe não efetuou a delineação antes da elaboração e somente um terço da classe não efetuou o *Brainstorming* antes da elaboração. É importante relatar também que houve 3 caminhos que foram criados somente por um grupo.



### 2.3.3 Discussão da Ferramenta

O trabalho abordado nessa seção é interessante, pois utiliza das informações dos processos das ferramentas Google Docs [5] e Glosser [24] para inferir um comportamento de um grupo. Nela, o autor não conseguiu chegar em uma conclusão de quais caminhos nos padrões de escrita levam a uma melhor escrita colaborativa. Porém, isso fornece material para impulsionar pesquisas futuras.

## 2.4 Discussão Final do Capítulo

Os trabalhos recentes sobre escrita colaborativa focam em ferramentas com características muito particulares, seja no aspecto técnico ou seja no aspecto teórico. Os trabalhos citados nesta seção foram escolhidos por estarem relacionados com a ferramenta de edição colaborativa Google Docs. Sendo de interesse da pesquisa do trabalho atual, eles serviram para dois aspectos importantes. O primeiro foi mostrar alguns limites e dados que poderiam ser utilizados e obtidos pela API. O segundo aspecto importante é a forma como foram estruturados os dados capturados da API.

## Capítulo 3

# Referencial Teórico: Pilares para Análise Semi-automática da Colaboração no Google Docs

Os itens abordados neste capítulo tratam dos elementos bases para o trabalho desenvolvido nesta monografia. O capítulo começa com a descrição de algumas conferências na área de estudos colaborativos. Em seguida, são feitas a apresentação do projeto engageGrid e engageGrid+, a apresentação da API do Google Docs diante de duas perspectivas e a apresentação de dois modelos de análise da colaboração. Por fim, o capítulo é encerrado com as considerações finais sobre as escolhas desses itens.

### 3.1 Colaboração Apoiada por Computador

Nesta seção é feita uma descrição da colaboração apoiada por computador. Para tal, é feita abordagem histórica e dos objetivos de algumas das conferências mais conhecidas na colaboração apoiada por computador: *Computer Supported Cooperative Work* (CSCW), *European Computer Supported Cooperative Work* (eCSCW), *Collaboration Researchers International Working Group* (CRIWG) e *Computer Suported Collaborative Learning* (CSCL). Por fim, será feita a descrição da área de sistemas colaborativos em âmbito nacional, representado pelo *Simpósio Brasileiro de Sistemas Colaborativos* (SBSC).

#### 3.1.1 CSCW

Historicamente, a relação entre homem e computadores em empresas começou por volta dos anos 60 com mainframes. Nos anos 70, as inovações tecnológicas na área da informática trouxeram novas formas de interação no ambiente de trabalho. Durante esse período, surgiu uma proposta de suporte a grupo de trabalho, chamado *Office Automation* [18], porém sem muito êxito. Foi somente nos anos 1984 que esses estudos criaram maturidade e surgiu a área conhecida como *Computer Supported Cooperative Work* (CSCW). Ela começou como workshop e em 1986 se tornou uma conferência de importância internacional. As conferências são organizadas pela *Association for Computer Machinery* (ACM) e atualmente está na sua décima quinta edição [3].

O CSCW é uma área interdisciplinar que tem como principal foco o desenvolvimento de tecnologia para o suporte à colaboração no trabalho e como o uso dessa tecnologia afeta a vida de pessoas, grupos e a sociedade. A conferência traz os maiores pesquisadores e praticantes da academia e da indústria que estão interessados tanto nos aspectos técnicos e sociais da colaboração.

### 3.1.2 eCSCW

O eCSCW possui o mesmo caráter do CSCW, porém ela ocorre no contexto europeu. Ela é promovida pela *European Society for Socially Embedded Technologies* (EUSSET). Apesar de seu foco ser parecido com o do CSCW, essa conferência aborda esse tema de acordo com as necessidades do panorama europeu, onde as pesquisas têm maior influência das nações, ao contrário dos EUA, onde a maior influência vêm das empresas. Podem-se destacar algumas questões abordadas pelo eCSCW: o papel da etnografia no CSCW e na computação em geral, o papel e a natureza da consciência no trabalho cooperativo, etc. O eCSCW nasceu no ano de 1989 e atualmente está na sua décima terceira [4] edição, realizando suas conferências bienalmente.

### 3.1.3 CRIWG

O *Collaboration Researchers International Working Group* (CRIWG) é uma comunidade aberta de pesquisadores. Ela promove a conferência em colaboração e tecnologia onde se pesquisa o projeto, desenvolvimento, implantação e uso de *groupware* e as práticas de trabalho que ele suporta.

Ele foi fundado em 1995 e se tornou um fórum significativo para pesquisadores da academia e profissionais para trocar ideias e experiências sobre problemas e soluções relacionadas com tecnologias de colaboração. Atualmente está na décima nona edição [1] e suas conferências ocorrem anualmente. A sua organização alterna locais entre as Américas e a Europa, com um número pequeno de apresentações, promovendo discussões, trabalho em equipe e a colaboração. O CRIWG se tornou um ícone internacional e interdisciplinar sobre tecnologia colaborativa e campos de aplicação relacionados, como CSCW. Ele desenvolve artigos sobre *groupwares* em uma larga variedade de perspectivas, variando da colaboração e pesquisa exploratória, para a construção de teorias e teste, pesquisa aplicada, engenharia e métodos de avaliação inovadores.

### 3.1.4 CSCL

O histórico de surgimento dessa área começa por volta de 1983 com um workshop intitulado *Joint problem solving and microcomputers*. O marco do nascimento do CSCL foi o workshop realizado em Maretea em 1989, pois foi a primeira vez que uma compilação pública e internacional empregou o termo *Computer Supported Collaborative Learning* em trabalhos. A primeira conferência realizada com o nome CSCL foi em 1995 e após esse ano foram realizadas reuniões bienais. Atualmente, essa conferência se encontra na decima edição [2].

Essa área tem como foco o estudo do aprendizado colaborativo com auxílio de computadores. O CSCL apresenta alguns desafios: a introdução da computação no aprendizado,

o aprendizado em grupos, o aprendizado individual em relação ao colaborativo, o papel da tecnologia no aprendizado, os processos de análise da colaboração, etc. Apesar de desenvolver ferramentas para colaboração no aprendizado, o foco do CSCL é no desenvolvimento de processos colaborativos de aprendizagem.

### 3.1.5 SBSC

O SBSC é um simpósio brasileiro [8] promovido pela Sociedade Brasileira de Computação (SBC) e discute o desenvolvimento e o uso de ferramentas no âmbito de sistemas colaborativos, com destaque para as redes sociais, wikis, sistemas de compartilhamento de arquivos, videoconferências, etc. Essas ferramentas desenvolvidas exploram várias áreas de estudo: Engenharia de Software, Banco de Dados, Inteligência Artificial, Sistemas de Informação, Computação Gráfica, Interação Humano-Computador, Sistemas Multimídia, Sistemas Distribuídos entre outras. As subáreas citadas anteriormente são todas disciplinas na área da computação e tratam da parte técnica do projeto. Ao tratar de questões não técnicas, outras áreas da ciência são abordadas: Sociologia, Antropologia, Psicologia, Educação, Linguística etc. O SBSC começou como um *workshop* com o nome de WCSCW em 2003, em 2005 seu nome foi alterado para SBSC e hoje já está na décima edição.

## 3.2 Fomento à Colaboração em Nuvem no Projeto EngageGrid+

Nessa seção é feita a descrição dos projetos EngageGrid e EngageGrid+ descrevendo suas propostas e objetivos.

### 3.2.1 EngageGrid

O projeto EngageGrid tem como objetivo prover suporte computacional que permita uma comunidade assumir compromissos em grupo de maneira fundamentada e segura. Para efetuar tal objetivo, foi desenvolvido um aparato conceitual, onde se definiu um ciclo de vida de um compromisso e houve sua prototipação em cima de um aparato tecnológico, como é descrito em “Rumo a uma rede de compromissos em *desktop* compartilhado: aspectos conceituais e tecnológicos” [14].

Um pressuposto dessa pesquisa é que as chances de compromissos bem sucedidos aumentam se eles são firmados de maneira bem fundamentada. Para tal, o aspecto conceitual desenvolve uma estrutura capaz de formalizar os compromissos. Essa estrutura é composta por 3 bases de dados e um ciclo dividido em 5 estágios que representam a vida de um compromisso no sistema. As bases de dados são caracterizadas abaixo:

- **Portfólio de competências:** É uma base de informações que contém as competências de cada integrante da rede.
- **Biblioteca de casos:** Apresenta um histórico com a descrição de todos os compromissos negados, aceitos e os executados com sucesso ou não.
- **Perfil da comunidade:** Apresenta informações que caracterizam o foco de trabalho de determinada comunidade e seu estado atual.

As bases de informações servem para alimentar uma primeira etapa de cinco do processo de gestão conhecido como ciclo de vida de um compromisso. Ao final dessas etapas, a biblioteca de casos é alimentada com informações do compromisso atual. Essas cinco etapas são descritas como:

- **Configuração:** Nessa fase, um indivíduo, caracterizado como proponente, efetua a criação da proposta de compromisso, seleção de integrantes, ferramentas, tempo de duração e definição do trabalho a ser executado. A base de informações presta suporte na escolha de indivíduos para projeto. O resultado dessa fase é um plano inicial de compromisso.
- **Negociação:** Nessa fase, o integrantes podem aceitar ou não o compromisso e têm a oportunidade de alterar a proposta e escolher um líder para o evento, que pode ser uma pessoa diferente do proponente.
- **Realização:** Essa fase é a responsável pela execução da tarefa puramente dita, onde pode haver alteração das ferramentas utilizadas, caso seja necessário, novos planos de entrada podem ser gerados (similar ao gerado na etapa de configuração). O produto dessa fase pode ser um ou mais.
- **Análise:** Essa fase realiza tanto uma análise explícita quanto implícita em relação a etapa anterior. A parte explícita ocorre através de questionários sobre o funcionamento do grupo e sobre o auto-desempenho. A avaliação implícita ocorre através de serviços dedicados da rede que realizam inferências sobre registros gerados na fase anterior. Ao final, é realizado o nível de alinhamento desse compromisso com o perfil da comunidade.
- **Síntese:** Essa fase é responsável por salvar os dados da fase de análise a fim de criar uma memória de casos. Essa memória irá compor a biblioteca de casos.

A figura 3.1 sintetiza a relação entre a base da dados e as etapas do ciclo de vida da rede compromissos.

O aparato tecnológico conta com a plataforma AGORA/GSD que serve de base para os serviços da rede de compromissos. Como descrito em [12], a escolha da plataforma AGORA foi devida a três características:

- **ubiquidade:** A implantação de recursos deve ser indiferente em relação a condições espaciais e temporais.
- **imanência:** É a capacidade de autonomia do grupo independente de seus integrantes não se conhecerem, permitindo a colaboração com confiança entre si mesmos.
- **Percepção:** Devido as diversas formas de comunicação, reforça o sentimento de presença do grupo.

Além dessas características, a plataforma é definida a partir de um modelo conceitual e serviços de base. O modelo conceitual é composto por 5 conceitos (agente, grupo, organização, recurso e atividade) ligados por 4 relações. Os serviços de base permitem a gestão do ambiente, que é construído pela instanciamento dos conceitos, e incluem autorização, notificação, gestão de membros, gestão de serviços, etc. Embutindo esses serviços, a plataforma oferece um serviço chamado *Grid Shared Desktop* (GSD).

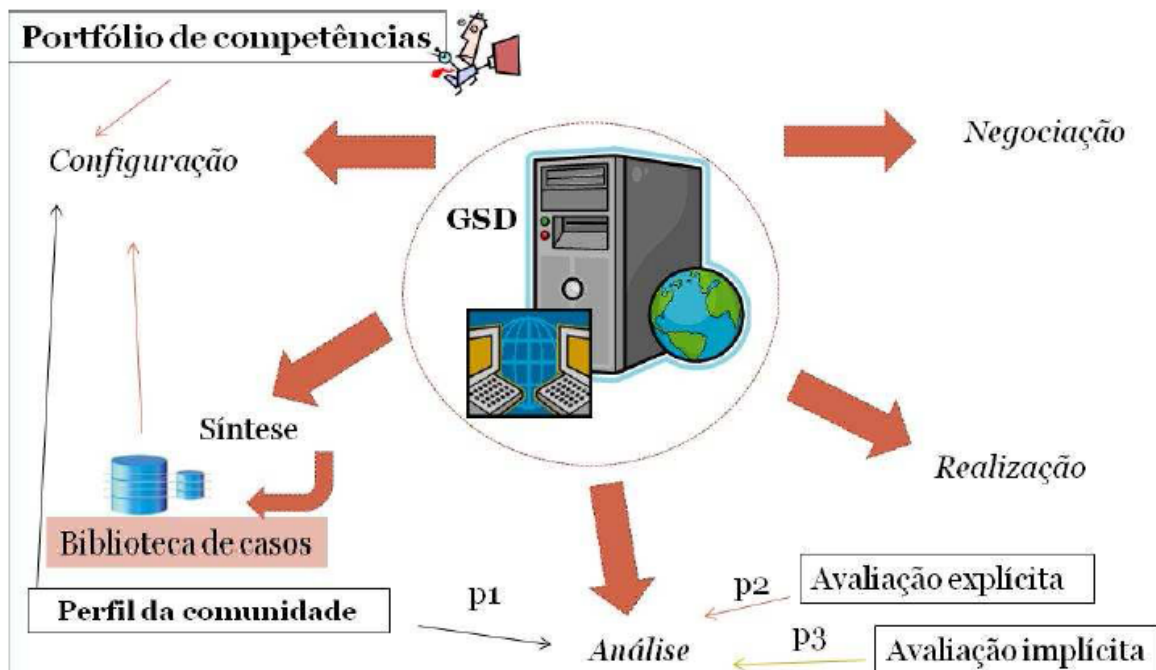


Figura 3.1: Etapas do ciclo de vida e sua relação com a base de dados. Fonte: [14].

O GSD apresenta compartilhamento de *desktops* para grupos de trabalho. Encontram-se nele dois níveis de colaboração: a comunidade virtual (CV) e as sessões colaborativas (SC). Na CV, ocorrem colaborações assíncronas através de serviços como o compartilhamento de arquivos, agenda de tarefas, edição de documentos, etc. Na SC, ocorrem colaborações síncronas como chat de texto, áudio e vídeo conferência, tela-branca (visualização do *desktop* atual por outros usuários da SC), quadro-branco (um *desktop* que pode ser manipulado pelos usuários da SC), etc.

### 3.2.2 EngageGrid+

Atualmente, os sistemas para colaboração via *web* se tornaram mais populares. Além do sistema AGORA/GSD descrito em [12], surgem outros como o GoogleDocs [5], Office Web Apps [7], etc. Esses sistemas utilizam a arquitetura em nuvem, disponibilizando o acesso a seus recursos através de um navegador *web*. Apesar de apresentarem muitas funções para o trabalho em grupo, eles não apresentam um sistema de gerenciamento de compromissos.

O projeto engageGrid+ [13] surgiu com a proposta do re-desenho dos serviços da rede compromissos a fim de manter um fraco acoplamento com a plataforma AGORA/GSD e dessa maneira disponibilizar a gestão de compromissos para outros sistemas. Mantendo o aparato conceitual criado no projeto engageGrid, o objetivo geral deste projeto é gerir compromissos em rede e em nuvem.

### 3.3 Google Docs: Capacidades e Limites

O Google Docs é uma ferramenta de colaboração que fornece serviços de edição, manipulação e armazenamento de documentos e arquivos. O seu sucesso é devido ao seu pioneirismo como uma ferramenta de edição simultânea de documentos via *web*. Essa seção tem como objetivo descrever as funcionalidades do Google Docs diante de duas perspectivas: do usuário e do desenvolvedor. Antes de descreve-las, alguns elementos presentes em ambas perspectivas serão descritos para melhor compreensão.

- **Documento:** É qualquer item mantido pelo Google Docs que é armazenado no formato nativo Google Docs. Estão inclusos documentos de texto, planilhas, apresentações ou desenhos. Um documento é criado ao criar um documento vazio ou ao fazer upload de um arquivo usando a interface *web* ou a API. Se esse arquivo for de um formato aceito pela API e o usuário optar por isto, ele se tornará um documento.
- **Arquivo:** É qualquer item mantido pelo Google Docs que é armazenado em um formato não nativo do Google Docs. Por exemplo, PDF, MPGE, MP3, etc. Arquivos que podem ser convertidos, se não o fizerem, serão tratados pela API como arquivos e perderão muitas das funcionalidades exclusivas para documentos.
- **Coleção:** É uma pasta que pode conter outras pastas, documentos e arquivos (Na versão atual, o termo pasta foi substituído por coleção).
- **Recurso:** Qualquer item mantido pela API, como por exemplo, documentos, arquivos e pastas.
- **Id do Recurso:** É um identificador único para cada recurso. Ele geralmente apresenta o formato *TIPO:ID*. Por exemplo, “document:12345”. É possível encontrar ids sem tipos, contendo somente o número.

#### 3.3.1 Perspectiva do Usuário

O Google Docs oferece acesso aos seus serviços a partir de uma interface *web*. Para um usuário ter acesso, basta ter uma conta registrada na Google, por exemplo, uma conta Gmail. Ao acessar o serviço, o Google disponibiliza duas telas de manipulação, a primeira permite a gerência e o compartilhamento de recursos, que nesse trabalho será chamada de tela de gerenciamento, e a segunda é responsável pela edição e visualização de documentos e arquivos, que nesse trabalho será chamada de tela de edição.

##### Tela de Gerenciamento

As estruturas presentes na tela de gerenciamento permitirão ao usuário acessar, compartilhar, deletar, buscar e baixar recursos. Também é possível alterar configurações da conta e da exibição de recursos. Os elementos presentes nessa tela são descritos a seguir (figura 3.2):

**A lista de recursos e botões relacionados:** A estrutura principal dessa interface é a lista de recursos, pois é ela que disponibiliza documentos, arquivos e pastas na conta atual. Todas as pesquisas efetuadas na interface, alteram os recursos exibidos nesse elemento.

Na lista, cada recurso apresenta informações importantes para o usuário como nome, criador, tipo (representado por um símbolo específico), data da última modificação e se está sendo compartilhado. Para o usuário poder manipular os documentos e arquivos, existe uma série de botões que estão logo acima da lista, cujas funções são ações de gerência tais como, abrir, deletar, compartilhar, favoritar, copiar, etc.

**As *labels*:** Elas são elementos que estão presentes no lado esquerdo da lista de recursos e abaixo dos botões de criação e *upload*. A sua função é servir como um filtro de pesquisa padronizado. Não é possível criar *labels* novas e para usa-las, basta um clique simples em seu nome e assim é efetuada uma pesquisa com o parâmetro selecionado.

**A barra de pesquisa:** Esse elemento permite o usuário fazer uma busca textual baseada nas informações presentes nos recursos da lista, como descrito no primeiro item. Por exemplo, pode-se procurar pela data, ou pelo nome do usuário criador, pelo título, etc.

**Os botões de criação e de *upload*:** O botão de criação permite a inserção de documentos (documentos de texto, desenhos, planilhas, etc) e pastas novos na lista de recursos. Recentemente, foi adicionada a funcionalidade que permite a utilização de aplicativos *web* para atuarem como manipuladores de arquivos não padrões, como arquivos de músicas, fotos, scripts, etc.

O botão de *upload* permite o usuário importar um documento para o Google Docs. Se esse documento for compatível com os tipos aceitos pela interface, ele será convertido para um formato padrão e poderá ser editado na tela de edição. Caso contrário, ele ficará armazenado e disponível para visualização e na maioria dos casos somente para *download*.

Existem algumas funcionalidades extras referentes a essa tela que são disponibilizadas toda vez que um arquivo externo é adicionado. Elas são o reconhecimento óptico de caracteres e a tradução de documento de texto.

## Tela de Edição

As estruturas presentes nessa interface são semelhantes às apresentadas em conhecidos softwares de edição de textos, como Microsoft Word ou Open Office. Elas serão descritas a seguir (Figura 3.3):

**A área de edição:** A área de edição é responsável pela edição em si do documento e pela visualização. Dependendo do tipo básico de documento que está sendo editado, a área de edição vai ter um formato específico, por exemplo, uma grade para planilhas ou um papel em branco para textos.

Nessa tela, os colaboradores realizam o trabalho cooperativo síncrono. Se dois colaboradores estiverem editando um documento ao mesmo tempo, será possível ver seus cursores de edição. Eles são diferenciados por cores que são atribuídas aos usuários no momento que acessam a tela de edição de determinado documento.



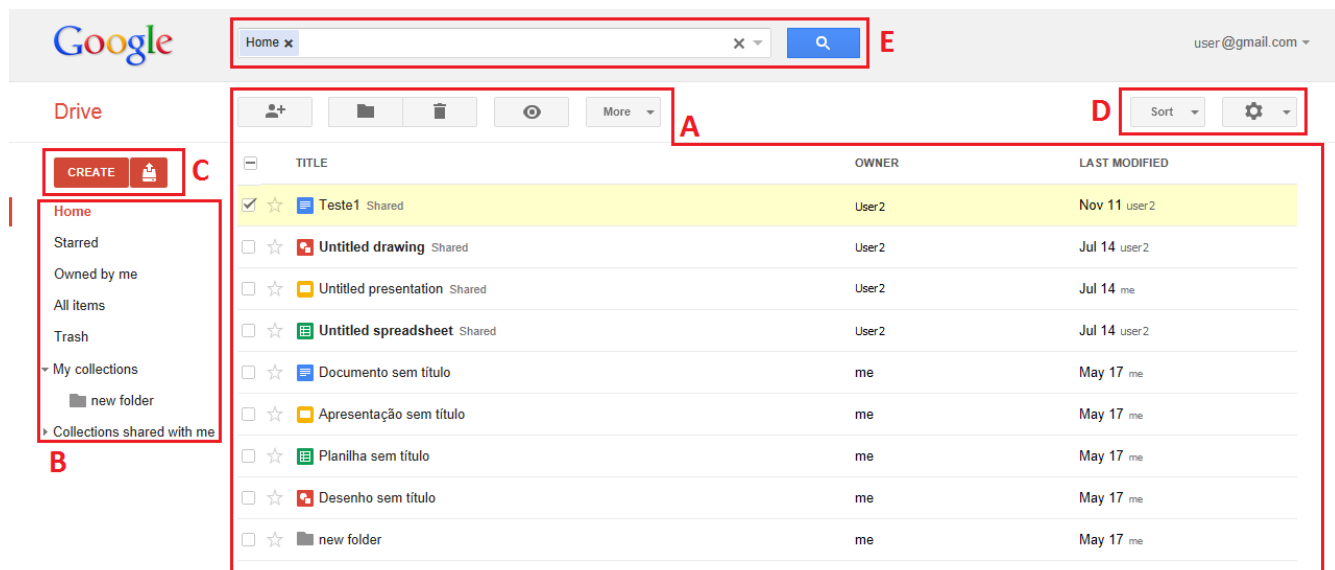


Figura 3.2: A tela da interface de gerência. (A) Lista de recursos. (B) *Labels*, (C) Botões de criar e *upload* de arquivos, (D) Botões de configuração (E) Barra de busca.

Alguns arquivos, como PDFs, podem ser visualizados, porém não poderão ser editados. Nos outros arquivos, que não são suportados pela ferramenta, a área de edição fica inoperante fornecendo somente a opção de *download* e verificação das revisões.

**A barra de ferramentas:** Esse elemento apresenta ferramentas importantes para editar o documento, por exemplo, tipo e tamanho da fonte, alinhamento do texto, etc.

**A barra de menus:** Ela apresenta as opções *Arquivo*, *Editar*, *Formatar*, *Ferramentas*, *Tabelas* e *Ajuda*. Através do campo arquivo, é possível abrir um documento, criar um documento novo, compartilhar, copiar, acessar o histórico de revisão e imprimir o documento. Os outros campos apresentam nomes autoexplicativos que não necessitam detalhamento.

**A barra de histórico e de *chat*:** No histórico, há uma lista, em que cada item mostra as seguintes informações, os colaboradores de determinada versão, a colaboração efetuada (esse item é mostrado diretamente na tela de edição) e data de seu salvamento. É possível ver o que cada integrante do grupo alterou em relação a uma outra versão pré-selecionada e também tornar como a mais recente uma versão passada.

O *chat* de texto aparece na parte direita da tela quando o documento está sendo compartilhado. Durante uma colaboração, cada membro apresenta um cor que o caracteriza no *chat* permitindo relaciona-los com os seus respectivos cursores de edição. O *chat* de texto não apresenta histórico, ou seja, após sair da tela de edição, o que foi digitado naquele *chat* é perdido.

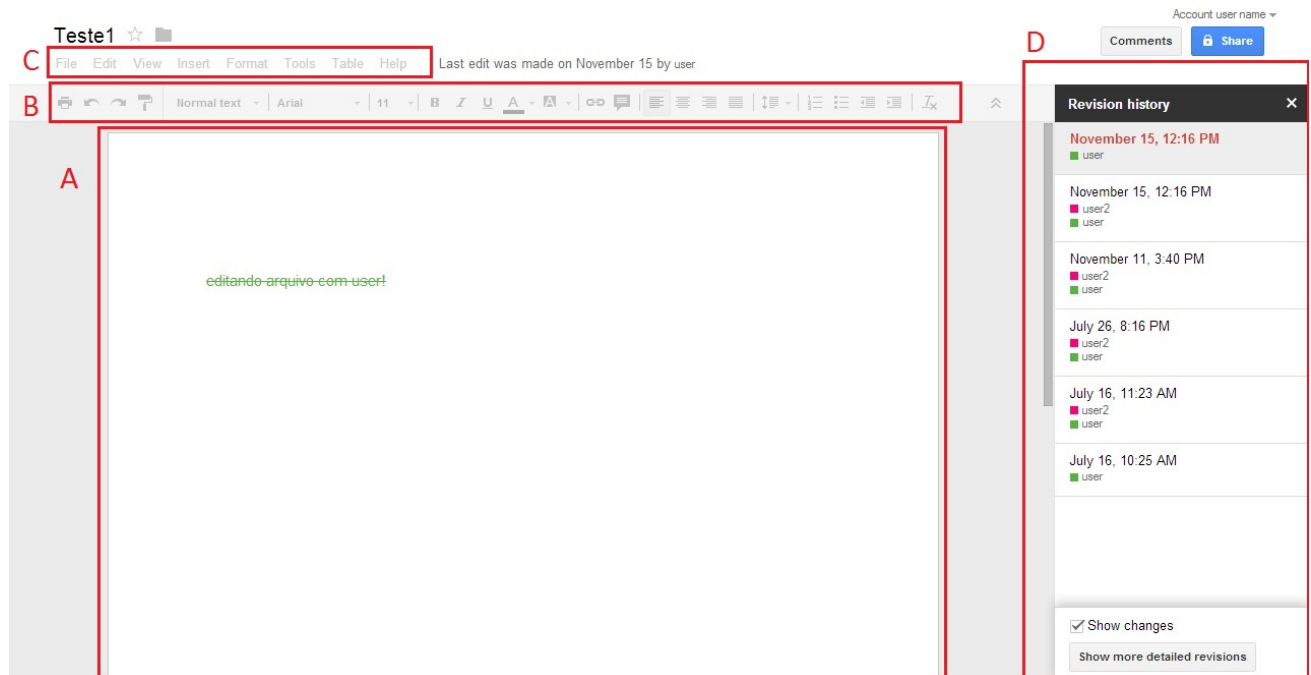


Figura 3.3: A tela da interface de Edição. (A) A área de edição, (B) a barra de ferramentas, (C) os menus e (D) a barra de revisão de histórico.

### 3.3.2 Perspectiva do Desenvolvedor

O Google Docs apresenta uma API que permite desenvolvedores customizarem os serviços existentes na plataforma *web*. A API descrita nesse trabalho se encontra na terceira versão [6] e com ela é possível criar, recuperar, atualizar e deletar documentos no formato do Google, arquivos e pastas. Ela também provê algumas funcionalidades como reconhecimento óptico de caracteres, tradução e histórico de revisão.

A API está disponível nas linguagens de programação Java e .NET. e a documentação apresenta informações sobre as requisições e respostas a nível de protocolo HTTP referentes às funções presentes nas linguagens suportadas. Os detalhes técnicos acerca das funcionalidades serão resumidos para melhor entendimento e para apresentar uma visão geral sem ligação com a sintaxe de alguma linguagem de programação.

#### *Feeds*

A forma de comunicação entre a aplicação e o servidor é através do protocolo HTTP. Algumas requisições têm como resposta listas (lista de alterações num documento, lista de revisões num arquivo, lista de permissão de acesso, etc) que são enviadas por meio de *Feeds* no formato ATOM<sup>1</sup>.

Os *feeds* são baseados em XML e possuem termos conhecidos como entradas (*entry*) que representam um elemento. Os subelementos presentes em uma entrada vão depender do tipo de lista de retorno que foi pedida na requisição. Por chamada, os *feeds* contêm,

<sup>1</sup>ATOM é um padrão baseado em XML que permite a recuperação de informações na internet na camada de aplicação.

cada um, 100 entradas, porém podem ser customizados para receberem menos ou mais no limite de 1000 entradas.

Abaixo segue um exemplo de entrada retornada em um *feed* de recursos:

---

```
<entry xmlns:gd="http://schemas.google.com/g/2005"
gd:etag="'HhJSFgpeRyt7ImBq'">
  <id>https://docs.google.com/feeds/id/pdf%3A12345</id>
  <published>2009-04-09T18:23:09.035Z</published>
  <updated>2009-04-09T18:23:09.035Z</updated>
  <app:edited
    xmlns:app="http://www.w3.org/2007/app">2009-06-18T22:16:02.388Z</app:edited>
  <category scheme="http://schemas.google.com/g/2005/labels"
    term="http://schemas.google.com/g/2005/labels#starred" label="starred"/>
  <category scheme="http://schemas.google.com/g/2005/labels"
    term="http://schemas.google.com/g/2005/labels#viewed" label="viewed"/>
  <category scheme="http://schemas.google.com/g/2005/labels"
    term="http://schemas.google.com/g/2005/labels#hidden" label="hidden"/>
  <category scheme="http://schemas.google.com/g/2005#kind"
    term="http://schemas.google.com/docs/2007#pdf" label="pdf"/>
  <title>PDFs Title</title>
  <content type="application/pdf"
    src="https://doc-04-20-docs.googleusercontent.com/docs/secure/m71240...U1?h=1630126&a
  <link rel="alternate" type="text/html"
    href="https://docs.google.com/fileview?id=12345&hl=en"/>
  <link rel="self" type="application/atom+xml"
    href="https://docs.google.com/feeds/default/private/full/pdf%3A12345"/>
  <link rel="edit" type="application/atom+xml"
    href="https://docs.google.com/feeds/default/private/full/pdf%3A12345"/>
  <link rel="edit-media" type="application/pdf"
    href="https://docs.google.com/feeds/default/media/pdf%3A12345"/>
  <link rel="http://schemas.google.com/g/2005#resumable-edit-media"
    type="application/atom+xml"
    href="https://docs.google.com/feeds/upload/create-session/default/private/full/pdf%3A1
  <author>
    <name>user</name>
    <email>user@gmail.com</email>
  </author>
  <gd:resourceId>pdf:12345</gd:resourceId>
  <gd:lastModifiedBy>
    <name>user</name>
    <email>user@gmail.com</email>
  </gd:lastModifiedBy>
  <gd:lastViewed>2009-06-18T22:16:02.384Z</gd:lastViewed>
  <gd:quotaBytesUsed>108538</gd:quotaBytesUsed>
  <docs:writersCanInvite value="false"/>
  <docs:md5Checksum>2b01142f7481c7b056c4b410d28f33cf</docs:md5Checksum>
  <gd:feedLink rel="http://schemas.google.com/acl/2007#accessControllist"
    href="https://docs.google.com/feeds/default/private/full/pdf%3A12345/acl"/>
```

```
<gd:feedLink rel="http://schemas.google.com/docs/2007/revisions"
  href="https://docs.google.com/feeds/default/private/full/document%3A12345/revisions"/>
</entry>
```

---

## Metadados do Usuário

A API permite que baixe metadados do usuário. Essa funcionalidade retorna dados sobre a conta de um determinado usuário que, por exemplo, podem ser usados para customizar uma interface gráfica da aplicação. Os dados são obtidos acessando as seguintes URLs <https://docs.google.com/feeds/metadata/<user-id>> e <https://docs.google.com/feeds/metadata/default>.

A primeira *url* é usada para buscar informações sobre um usuário qualquer onde “<user-id>” é a conta de email cadastrado no Google e a segunda busca os dados do atual usuário autenticado na aplicação. A resposta a essa *url* é um *feed* com a propriedade de somente leitura contendo somente uma entrada que possui informações públicas do usuário.

## Autorização de Requisições

A API do Google Docs trabalha em parte com dados não públicos e por isso é necessário fazer uma autorização das requisições. Ela permite utilizar alguns protocolos de autorização como o OAuth 1.0, OAuth 2.0, AuthSub e Client Login.

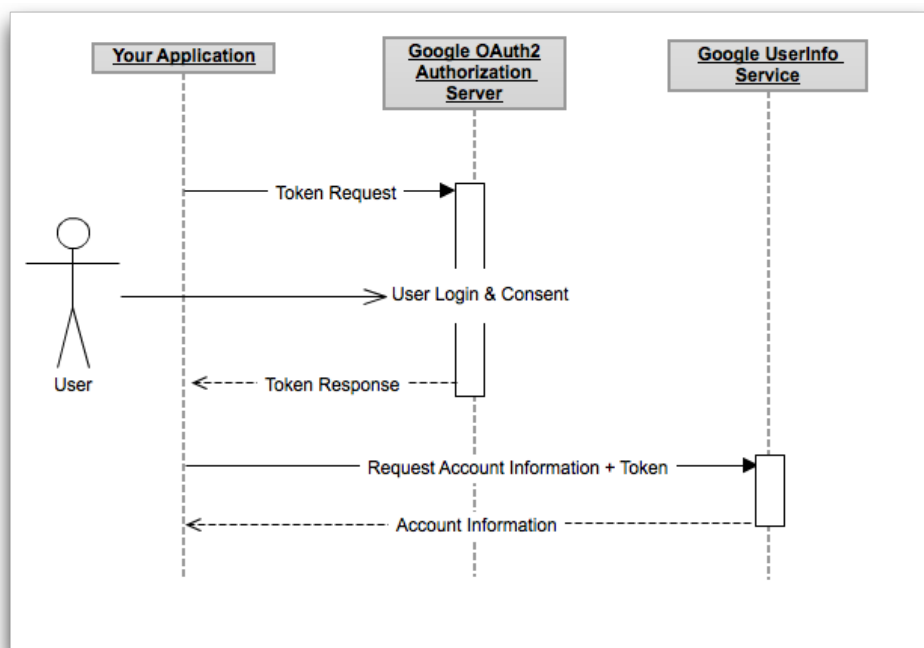


Figura 3.4: Diagrama de sequência para autorização usando OAuth 2.0. Fonte:<https://developers.google.com/accounts/docs/OAuth2?hl=pt-BR>.

A função dos protocolos como OAuth 1.0, 2.0 e AuthSub é criar uma forma de terceiros poderem acessar dados de um usuário sem ter a necessidade de manipular as suas creden-

ciais (nome do usuário e senha), garantindo assim um escopo de acesso para a aplicação. Para tal, é necessário cadastramento dessa aplicação perante o Google, o que garante um identificador à aplicação.

A figura 3.4 mostra o diagrama de sequência da autenticação utilizando o OAuth 2.0. No momento que um usuário autoriza algum aplicativo acessar seus dados, o Google fornece um *token* para a aplicação e a partir dele será feita a identificação de qual aplicação pode acessar quais dados de um determinado usuário.

Diferente das outras formas de autorização de requisição, o ClientLogin utiliza um método cliente-servidor de autorização, por meio do fornecimento direto das credenciais do usuário. Dessa maneira, esse método não é recomendado, pois o vazamento de informações podem ser muito nocivas para o usuário.

## Detecção de Alterações em Recursos

A detecção de alterações funciona através de um *feed*, conhecido como *feed* de alterações. Ele fornece informações sobre todas alterações em todos recursos de uma conta, inclusive os compartilhados. O *feed* utiliza um elemento chamado etiqueta de mudança (*changestamp*). Ele é definido como um inteiro único que aumenta monotonicamente representando uma alteração a um recurso da lista de documentos de um usuário.

O *feed* fornece o estado atual dos recursos se, e somente se o recurso foi alterado desde de uma etiqueta de mudança. Portanto, ele fornece um meio eficaz da detecção de mudança quando comparado ao método anterior, onde todos os recursos do usuário eram verificados repetidamente.

O procedimento básico para checagem de alterações nos recursos consiste em armazenar o valor da etiqueta de mudança salva pela aplicação e comparar com a que está presente. Esse método evita que se recupere o mesmo recurso mais de uma vez.

Segue abaixo, o exemplo de uma entrada de um *feed* de alterações representando a criação de um documento.

---

```
<entry gd:etag="\"BIAIEQdOHct7ImBr\"">
  <id>https://docs.google.com/feeds/id/document%3A1DJ1Qrt_N41aX1nDxxo0QHs5XIM90zTT-seYG2FYx
  <published>2011-06-13T15:05:37.544Z</published>
  <updated>2011-06-13T15:05:38.108Z</updated>
  <app:edited
    xmlns:app="http://www.w3.org/2007/app">2011-06-13T15:05:38.456Z</app:edited>
  <category scheme="http://schemas.google.com/g/2005#kind"
    term="http://schemas.google.com/docs/2007#document" label="document"/>
  <category scheme="http://schemas.google.com/g/2005/labels"
    term="http://schemas.google.com/g/2005/labels#viewed" label="viewed"/>
  <title>Meeting Notes</title>
  <content type="text/html"
    src="https://docs.google.com/feeds/download/documents/export/Export?id=1DJ1Qrt_N41aX1n
  <link rel="alternate" type="text/html"
    href="https://docs.google.com/a/example.com/document/d/1DJ1Qrt_N41aX1nDxxo0QHs5XIM90zT
  <link rel="http://schemas.google.com/g/2005#resumable-edit-media"
    type="application/atom+xml"
    href="https://docs.google.com/feeds/upload/create-session/default/private/full/documen
```

```

<link rel="http://schemas.google.com/docs/2007/thumbnail" type="image/jpeg"
      href="https://lh5.googleusercontent.com/u7NNUBIcKwb74epx_cb5ff_UoJDo5wm5M9zVRZq33qDfTn
<link rel="edit" type="application/atom+xml"
      href="https://docs.google.com/feeds/default/private/full/document%3A1DJ1Qrt_N41aX1nDxx
<link rel="edit-media" type="text/html"
      href="https://docs.google.com/feeds/default/media/document%3A1DJ1Qrt_N41aX1nDxxo0QHs5X
<link rel="self" type="application/atom+xml"
      href="https://docs.google.com/feeds/default/private/changes/5635"/>
<author>
  <name>joe.smith</name>
  <email>joe.smith@example.com</email>
</author>
<gd:resourceId>document:1DJ1Qrt_N41aX1nDxxo0QHs5XIM9OzTT-seYG2FYxrRk</gd:resourceId>
<gd:lastModifiedBy>
  <name>joe.smith</name>
  <email>joe.smith@example.com</email>
</gd:lastModifiedBy>
<gd:lastViewed>2011-06-13T15:05:38.108Z</gd:lastViewed>
<gd:quotaBytesUsed>0</gd:quotaBytesUsed>
<docs:writersCanInvite value="true"/>
<docs:changestamp value="5635"/>
<gd:feedLink rel="http://schemas.google.com/acl/2007#accessControllist"
      href="https://docs.google.com/feeds/default/private/full/document%3A1DJ1Qrt_N41aX1nDxx
<gd:feedLink rel="http://schemas.google.com/docs/2007/revisions"
      href="https://docs.google.com/feeds/default/private/full/document%3A1DJ1Qrt_N41aX1nDxx
</entry>

```

---

## Gerenciamento de Conteúdo e de Metadados de Documentos e Arquivos

O gerenciamento de documentos e arquivos se caracteriza pelas seguintes ações:

- Pesquisa de documentos, arquivos e coleções;
- A criação, *upload* e *download* de documentos e arquivos;
- A atualização de documentos e arquivos;
- A gerência de revisões;
- A eliminação e o movimento para a lixeira;
- Gerenciamento de ficheiros de documentos e arquivos.

Através da seguinte *URL* <https://docs.google.com/feeds/default/private/full>, recupera-se o *feed* com as entradas de todos os recursos e por meio de algumas de suas variações, são feitas as ações de gerência.

**Pesquisa** A pesquisa é feita pela recuperação do *feed* de documentos, arquivos e coleções com uso de parâmetros. Por exemplo, ao fazer a requisição da lista, basta acrescentar algum parâmetro na *url*, como */-/document*, para retorna-la somente com documentos

de texto. Ela permite a seleção de recursos por título, tipo, recursos deletados, estrelados (favoritados), além do uso do parâmetro “q” que permite a aplicação pesquisar por palavras, conjunções de palavras, disjunções e exclusões no título do documento.

**Criação e Upload** A criação/*upload* de recursos (exceto pastas) é a forma como recursos entram numa conta. O *upload* apresenta um sistema conhecido como *upload* com possibilidade de retorno, que funciona através do particionamento dos recursos em vários pedaços, permitindo o reenvio de partes que falharam ao serem enviadas. Dentre as vantagens desse *upload*, destacam-se a facilidade para o envio de arquivos grandes, a melhor performance em ambientes com largura de banda restrita, etc.

Durante a criação ou *upload*, o desenvolvedor se encontrará em uns desses três cenários: criar um documento ou arquivo somente com metadados, fazer *upload* de um documento ou arquivo somente com conteúdo e fazer *upload* de um documento ou arquivo com conteúdo e metadados.

Ao realizar o *upload*, ainda existe a possibilidade de traduzir o documento ou usar a leitura óptica de caracteres. A primeira funcionalidade exige que se escolha uma língua para poder realizar a tradução. A segunda funcionalidade recebe um PDF ou uma imagem com caracteres legíveis e os transforma em um documento de texto.

**Download** O *download* de recursos pode ser feito a partir do uso da lista de recursos. A partir de uma entrada da lista, a aplicação obtém o *link* contendo *URL* para o *download*. Essa *URL* deverá conter informações como o tipo do documento, o id e qual será o formato que o documento deverá ter ao ser baixado. Esse processo descrito também é usado para arquivos, exceto pelo fato da *URL* não conter o tipo e nem o formato para *download*.

O formato do documento ao ser baixado por padrão é HTML, entretanto, outros formatos podem ser usados para a *download* do documento. Para arquivos, o formato padrão é o mesmo usado no momento do *upload*. Segue abaixo os formatos possíveis para *download* de documentos.

- **Documento de texto:** docx, doc, html, jpeg, odt, pdf, png, rtf, txt e zip.
- **Desenho:** jpeg, pdf, png e svg.
- **Apresentação:** pdf, png, ppt e txt.
- **Planilha:** xlsx, xls, csv, pdf, ods, tsv e html.

**Atualização** A API permite atualização dos recursos presentes na lista. As possibilidades de atualização são os metadados do recurso, o conteúdo de um documento ou ambos numa mesma requisição. As atualizações são efetuadas através do uso de um *link* chamado *media*, para atualizar metadados, e *edit-media*, para atualizar conteúdo, presentes numa entrada de um recurso na lista.

Para atualizar os metadados, é necessário somente enviar um PUT para o *link* citado anteriormente e alterar os dados da entrada. Para efetuar uma atualização de conteúdo, envia-se um arquivo contendo as alterações, sendo esse arquivo compatível para *upload*.

Segue abaixo um exemplo de entrada da lista de recursos mostrando somente os *links* citados na atualização.

---

```
<entry xmlns:gd="http://schemas.google.com/g/2005" gd:etag="'HhJSFgpeRyt7ImBq'">
  ...
  <title>PDFs Title</title>
  ...
  <link rel="edit" type="application/atom+xml"
    href="https://docs.google.com/feeds/default/private/full/pdf%3A12345"/>
  <link rel="edit-media" type="application/pdf"
    href="https://docs.google.com/feeds/default/media/pdf%3A12345"/>
  ...
</entry>
```

---

**Histórico de Revisão** Essa API disponibiliza a funcionalidade do histórico de revisão. Ele é acessado através da URL [https://docs.google.com/feeds/default/private/full/resource\\_id/revisions](https://docs.google.com/feeds/default/private/full/resource_id/revisions) que retorna um *feed* de revisões e todos arquivos e documentos apresentam um *link* para ele. Cada entrada do *link* apresenta alguns dados como a data de atualização, autor da revisão e se o documento está publicado. As operações com revisões são a publicação, a deleção e o *download*.

Segue abaixo um exemplo de entrada do *feed* de revisões.

---

```
<entry>
  <id>https://docs.google.com/feeds/id/resource_id/revisions/0</id>
  <updated>2009-08-17T04:22:10.440Z</updated>
  <app:edited
    xmlns:app="http://www.w3.org/2007/app">2009-08-06T03:25:07.798Z</app:edited>
  <title>Revision 0</title>
  <content type="text/html"
    src="https://docs.google.com/feeds/download/documents/Export?docId=doc_id&revision=0"
  <link rel="alternate" type="text/html"
    href="https://docs.google.com/Doc?id=doc_id&revision=0"/>
  <link rel="self" type="application/atom+xml"
    href="https://docs.google.com/feeds/default/private/full/resource_id/revisions/0"/>
  <author>
    <name>user</name>
    <email>user@gmail.com</email>
  </author>
</entry>
```

---

É importante esclarecer um item da API que não está presente na documentação. Esse item é a definição explícita do que é uma unidade de revisão, pois esse termo pode apresentar algumas diferenças entre o que é visto na API e o que aparece na interface do usuário.

Ao se acessar a tela de edição, a interface do usuário recebe dados no formato de texto estruturado no padrão JSON. Esses dados contêm todas as informações do que vai aparecer no editor. Através deles é possível ver o que é considerado uma unidade de revisão.



Uma unidade de revisão extraída desses dados contém comandos que devem ser feitos para montar o texto. Esses comandos podem ser por exemplo, de inserção, deleção, edição, substituição, etc. Cada unidade de revisão pode conter um ou mais comandos desses citados. A quantidade não é padronizada, podendo ter no mínimo somente um. Após a série de comandos, aparece os seguinte dados, uma etiqueta de tempo em milissegundos, o id do usuário, o número da revisão e o id da sessão em que foram feitas as modificações. Essa descrição se refere a uma unidade de revisão. Abaixo segue alguns dos comandos que podem estar presentes numa unidade de revisão e depois é exibido os campos referentes ao id do usuário, sessão, etc.

- **Inserção:**

---

```
{  
  "ty": "is",  
  "s": "e",  
  "ibi": 2  
}
```

---

O campo “ty” representa o comando que vai ser executado. O valor “is” representa a ação de inserção. O campo “s” representa o que vai ser inserido no documento, nesse caso o caractere “e”. O campo “ibi” representa em que posição do documento o caractere vai ser inserido, nesse caso, a posição 2.

- **Deleção:**

---

```
{  
  "ty": "ds",  
  "ei": 25,  
  "si": 1  
}
```

---

Similar ao primeiro item, “ds” simboliza o comando de deleção. Os campos “ei” e “si” representam, respectivamente, a localização do último caractere a ser deletado e o primeiro caractere que deve ser deletado, nessa faixa de valores (todos os valores entre 1 e 25).

Cada revisão é identificada pela seguinte estrutura final em uma unidade do *changelog*.

---

```
[comandos]  
,  
1373646711942,           //TIMESTAMP  
"17356221186718672231", //ID USUARIO  
33,                     //NUMERO DA REVISAO  
"442ecaf63ed25376",    //NUMERO DA SESSAO  
31                      //NUMERO DO ITEM NA LISTA DO CHANGELOG
```

---

Na interface do usuário, são apresentadas uma quantidade bem menor de opções de revisão para serem selecionadas. Isso acontece, pois essas versões são condensadas para

economizar espaço no servidor. Ou seja, uma revisão mostrada na interface do usuário são na verdade várias unidades de revisão juntas.

**Deleção e Mover para a Lixeira** O processo de deleção e mover para a lixeira são bem parecidos. Eles utilizam uma requisição DELETE do HTTP com a mesma URL para o *update*. A diferença entre eles consta no fato de que a remoção move o recurso para a lixeira e a deleção elimina de vez o recurso de determinada conta. A chamada de deleção se difere da chamada de remoção ao acrescentar o seguinte campo, `delete=true`, ao fim da URL já citada anteriormente.

## Gerenciamento de Coleções e seus Conteúdos

As coleções presentes no Google Docs têm uma relação entre seus recursos onde eles podem existir em mais de uma coleção ao mesmo tempo. Cada recurso possui um grupo de *links* pais que apontam para coleções. Um *link* pai não pode aparecer mais de uma vez em um grupo de *links* e um grupo de *links* pode ter zero ou mais *links* pais. As operações existentes para coleções são obter lista de coleções, obter lista de recursos presentes na coleção raiz, criar e remover coleções, adicionar e remover recursos de coleções e deletar e mover coleção para lixeira.

As operações descritas anteriormente seguem o mesmo procedimento apresentado na sessão de gerenciamento de documentos, fazendo as devidas alterações nas URL de requisição.

## Gerenciamento de Permissões de Recursos via Access Control List (ACL)

ACL são listas de controle de acesso que são vinculadas a cada documento e pasta. Elas determinam quem tem acesso a um recurso. Nos *feeds* de acesso, são encontrados papéis disponíveis para um documento ou pasta:

- **Proprietário do recurso:** É o dono do documento. Ele tem direito de alterar a ACL desses recurso e outras ações, como deletar.
- **Escritor:** Esse papel permite a edição do documento.
- **Leitor:** Esse papel permite somente a leitura do documento.

Além dos papéis, existem níveis de compartilhamento:

- **Usuário:** Recurso vinculado a um email.
- **Grupo:** Recurso vinculado a um grupo de emails da Google.
- **Domínio:** Recurso vinculado a um domínio da Google Apps.
- **Padrão:** Recurso compartilhado com todos os usuários.

As operações sobre ACLs são a recuperação de ACL para um determinado documento, arquivo ou pasta, atualização e remoção das permissões de compartilhamento, email de notificação em mudanças de ACL e o compartilhamento de recursos utilizando chaves.

## Gerenciamento de Ficheiros de Documentos e Arquivos

Essa última opção de gerência de recurso permite a aplicação exportar documentos e arquivos no formato .ZIP. Para efetuar essa operação, é necessário fazer um requisição POST com a URL <https://docs.google.com/feeds/default/private/archive> e nela colocar informações como os ids dos recursos, seus formatos de exportação e uma lista de emails para notificação. As operações para gerência de ficheiros são criação de ficheiros, obter status de ficheiros, atualizar a lista de emails de notificação e cancelar a criação de um ficheiro.

Segue abaixo um exemplo de *feed* de criação de ficheiro.

---

```
POST https://docs.google.com/feeds/default/private/archive
Authorization: <your authorization header here>
Content-Length: 457
Content-Type: application/atom+xml

<entry xmlns="http://www.w3.org/2005/Atom"
  xmlns:docs="http://schemas.google.com/docs/2007">
  <docs:archiveResourceId>document:0A1234567890</docs:archiveResourceId>
  <docs:archiveResourceId>spreadsheet:0A1234567890</docs:archiveResourceId>
  <docs:archiveResourceId>presentation:0A1234567890</docs:archiveResourceId>
  <docs:archiveResourceId>pdf:0A12sdf</docs:archiveResourceId>
  <docs:archiveResourceId>folder:0A1234567890</docs:archiveResourceId>
  <docs:archiveConversion source='application/vnd.google-apps.document'
    target='application/msword'/>
  <docs:archiveConversion source='application/vnd.google-apps.spreadsheet'
    target='text/csv'/>
  <docs:archiveConversion source='application/pdf' target='application/pdf'/>
  <docs:archiveNotify>user@gmail.com</docs:archiveNotify>
</entry>
```

---

## 3.4 Modelos para Análise da Colaboração

Nessa seção, é realizada a descrição de dois modelos para análise da colaboração. O primeiro é um sistema de aprendizagem DEGREE, que tem como objetivo medir o grau de colaboração de grupos e indivíduos. A sua estrutura será detalhada descrevendo todo o processo de colaboração e a sua forma de comunicação. Ao final é descrita sua análise.

O segundo é o modelo 3C, que é usado para medir o grau de colaboração de sistemas colaborativos, principalmente na fase de criação desses *groupwares*. Ele divide o elemento colaboração em 3 outros, a saber, cooperação, coordenação e comunicação e assim é feita a análise de cada elemento e as suas dependências.

### 3.4.1 Modelo de Análise do Sistema DEGREE

O DEGREE [10] é um sistema colaborativo que tem como objetivo a análise comportamental de grupos e seus integrantes. O funcionamento do sistema é baseado em

atividades de aprendizagem criadas por tutores onde aprendizes devem colaborar para solucioná-las.

O sistema recolhe várias informações obtidas durante algumas etapas pré-estabelecidas e caracteriza o comportamento num conjunto de atributos. A partir desse conjunto de atributos, é feita uma análise em 3 perspectivas: o desempenho de um grupo em relação a outros, o desempenho de um membro em relação a outros e o desempenho do grupo e relação a si mesmo. A colaboração durante a atividade proposta é feita através de um sistema de mensagens semi-estruturadas que servirão também como fonte de dados para a análise. O sistema ainda possui ferramentas para criação das atividades, análise e extração dos dados.

Alguns princípios básicos do CSCL [2] são abordados nesse sistema: O construção conjunta da solução de problemas, a coordenação dos membros de um grupo para o planejamento de tarefas, a semi-estruturação de mecanismos de interação e foco tanto no processo de aprendizagem quanto no resultado.

### Organização do Sistema

O sistema propõe o conceito de *atividade* como unidade básica para representação de atividades onde a tecnologia tem o papel de mediador. Ela é constituída de alguns elementos (veja na figura 3.5). Esses elementos são: objetivos, tema, comunidade envolvida, regras, ferramentas, a divisão de trabalho e o resultado.

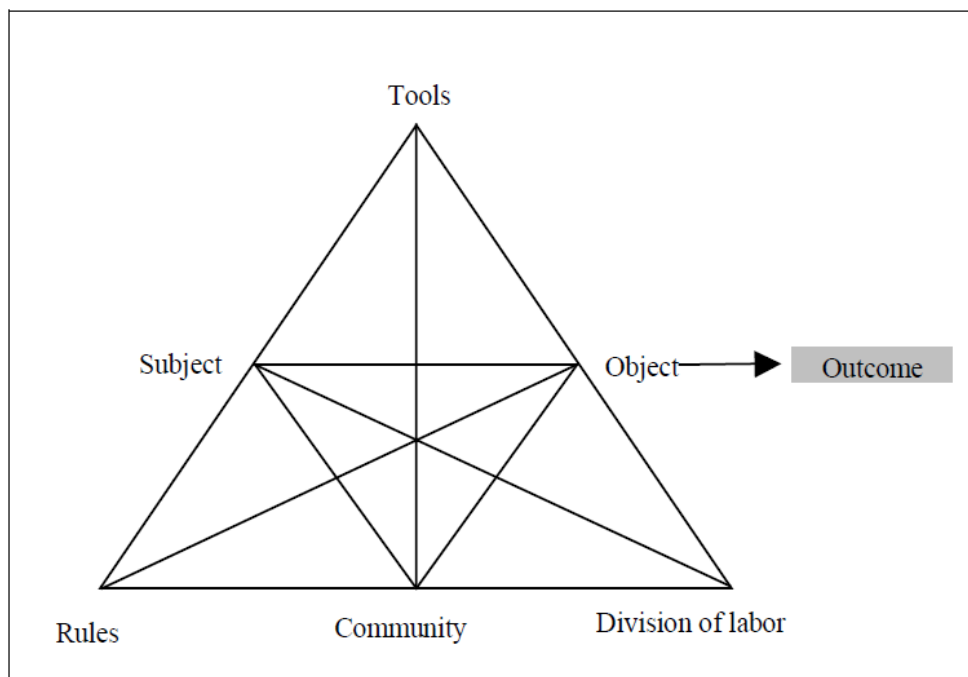


Figura 3.5: Definição do conceito de atividade. Fonte: [10].

A arquitetura do sistema se organiza em 4 níveis:

- **Configuração:** Com as atividade definidas, os tutores configuram e instalam o ambiente que servirá de suporte para os grupos de estudantes. Esse ambiente vai

fornecer recursos necessários para execução das tarefas conjuntas. Nessa etapa os tutores definem tarefas, recursos e grupos partindo do zero ou reutilizando componentes de outras atividades já realizadas.

- **Performance:** Nessa etapa, os estudantes realizam as atividades criadas na fase de configuração com suporte do sistema. As atividades envolvem uma série de tarefas em áreas de trabalho compartilhadas. O sistema é baseado em conversação. Além disso, ele gerencia as colaborações dos usuários, nomeando as contribuições, dando suporte para a co-construção da solução num processo de discussão argumentativo colaborativo. Todo esse processo é salvo e pode ser analisado ou reusado para diferentes propostas nas etapas de análise e organização.
- **Análise:** Nessa etapa, é feita a análise da interação do usuário e também são feitas intervenções para melhorar seu desempenho. Há ferramentas de análise quantitativa e qualitativa para analisar e observar a solução das tarefas da etapa de performance. Nessa etapa, é proposto um modo de observar e dar valores para as atitudes dos usuários quando eles estão trabalhando em conjunto. Existe a possibilidade de enviar mensagens para o grupo ou para os indivíduos, informando modos para melhorarem os seus desempenhos nas atividades. Finalmente são registradas as mensagens e o momento em que são feitas as intervenções e análise das melhorias.
- **Organização:** Nessa etapa, ocorre a seleção e armazenamento das experiências de aprendizado colaborativo. A informação é estruturada e valorada para pesquisa e reuso. Ela também é armazenada em casos para formar uma memória de aprendizado organizacional.

Unindo a arquitetura descrita anteriormente com os elementos presentes no conceito de *atividade*, o sistema apresenta 4 atividades (o processo das 4 atividades é conhecido como experiência). As atividades são descritas a seguir:

- **Configuração de uma experiência:** Um ou mais professores (A comunidade) definem algumas metas de aprendizado (o objetivo) que serão procedidas para configurar uma experiência de aprendizado (resultado). O subsistema do nível de configuração disponibiliza o suporte para a criação dos componentes para a experiência. Casos anteriores também podem ser usados como ferramentas para configuração. A experiência é definida por um processo de quatro etapas: a definição do grupo, a definição da estrutura da conversação, a definição do espaço de trabalho compartilhado e as atividades que integrarão a experiência. A definição da experiência e o suporte ao ambiente servirão de ferramentas para outras etapas.
- **Execução uma experiência:** Um grupo de estudantes e possivelmente um grupo de professores (comunidade) executam uma experiência através de um *framework* computacional (ferramentas) adaptado para o trabalho colaborativo a distância para alcançar algumas metas comuns de aprendizado. Uma experiência pode ser dividida em tarefas e subtarefas (divisão das atividades). Os estudantes aceitam participar e compartilhar a responsabilidade do processo e seus resultados assumindo um papel particular e utilizando as regras de conversação para a discussão argumentativa. A tarefa é considerada acabada quando existir um documento juntamente elaborado e aceitado pelo grupo. O resultado dessa experiência é um documento para cada

tarefa assim como uma representação do processo de elaboração. O resultado dessa atividade será usado em outras etapas, particularmente na “gerência de casos” e “análise de uma experiência”.

- **Análise de uma experiência:** Essa atividade geralmente envolve professores procurando pelas ações e interações de estudantes enquanto eles executam a experiência de aprendizado (comunidade). As ferramentas usadas para a execução da atividade são, por um lado, o ambiente de análise DEGREE, provendo ferramentas gráficas e métodos de inferência, e por outro lado, todo o dado sobre a definição da experiência incluída no nível de configuração. A tarefa de análise é feita automaticamente ao aplicar as regras de avaliação qualitativas, enquanto o alerta e o monitoramento são suportados pelo sistema, porém é realizado sob a vontade do tutor. O objeto de análise são os dados recolhidos na etapa de execução, incluindo toda as informações de ações e interações realizadas pelos participantes. O resultado dessa atividade vai ser usado na etapa final, a Gerência de casos, e servirá de *feedback* para outras experiências futuras para melhorar as atitudes dos estudantes.
- **Gerência de casos para formar uma memória de casos organizacional:** Os professores (comunidade) criam e adicionam casos das experiências executadas pelo sistema. Para fazê-lo, é utilizada uma ferramenta do DEGREE que permite a visualização, seleção, anotação e avaliação dos resultados e processos para criar novos casos. O resultado dessa etapa é a biblioteca de casos, que pode ser utilizada para definir novas experiências.

O modelo da estrutura de conversação mencionado na etapa de configuração de uma atividade é um grafo orientado etiquetado. Esse mecanismo permite a estruturação da conversa de um grupo de forma genérica. Os nós representam o tipo de contribuição e as pontas representam as ligações entre contribuições. Os nós podem ser, por exemplo, “Proposta”, “Contraproposta”, “Pergunta”, “Comentário” e “Acordo”, como pode ser visto na figura 3.6.

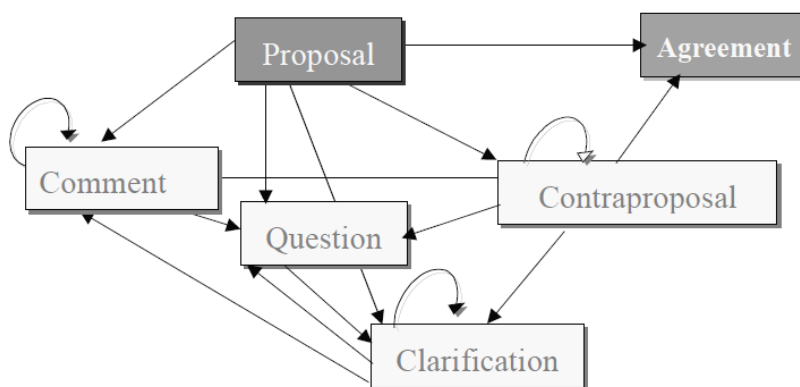


Figura 3.6: Um exemplo de conversa semi-estruturada. Fonte: [10].

A estrutura da conversação é algo a ser definido pelos tutores na fase de configuração. Para haver um análise da interação, para cada tipo de contribuição, foi definido uma

valoração (de 10 a -10) num conjunto de atributos. Eles são: iniciativa, criatividade, elaboração e conformidade.

Durante a fase de execução, os alunos têm acesso à interface de colaboração. Nela é possível a troca de mensagens da conversa semi-estruturada. Ela é basicamente montada com opções do tipo de mensagem, por exemplo, “proposta”, e apresenta um campo com o conteúdo da mensagem. Esse modo de conversação cria uma sequência de mensagens que formam uma estrutura em árvore, onde mais a frente será usada na análise.

## Análise da Interação

Durante o processo de resolução das tarefas, o sistema salva todas as informações importantes dos usuários. As informações registradas são: a identificação do usuário, o horário e a data, o computador host, a experiência de aprendizado, o grupo, a atividade, a tarefa, o espaço de trabalho e o tipo de ação. Essas informações são salvas em um banco de dados relacional e uma pesquisa com uma combinação de critérios podem facilmente retornar os dados desejados. Com essas facilidades, a execução da análise pode ser feita de três modos.

Os modos de análise são os seguintes:

- **Análise Quantitativa:** Pesquisas a esse módulo abrangem uma variedade de parâmetros e a saída pode ser representada de forma textual ou gráfica.
- **Análise Qualitativa:** Considera a atividade de configuração de uma experiência, o processo e o resultado, mais o conhecimento subjetivo de um observador sobre a interpretação dos dados observados. Essa análise aborda três dimensões: Um grupo em relação a outro, um indivíduo em relação a outro do mesmo grupo e o grupo por si mesmo. Nessa análise, também é feito o envio de *feedback* do trabalho realizado.
- **Feedback da Análise:** usa a saída do sistema de avisos e graficamente representa a evolução de um grupo de acordo com o subconjunto de variáveis observadas.

## Análise Quantitativa

Uma interface baseada na *web* é fornecida para a seleção de diferentes opções para pesquisa, onde os dados podem ser exibidos de forma textual ou gráfica. É possível visualizar a interação do usuário de diferentes modos:

- Evolução do número de contribuições de usuários numa experiência durante um período, usando uma exibição gráfica;
- Plotar o número de acessos por hora para um grupo numa atividade;
- Número de contribuições por usuário, para todos os usuários de um grupo;
- Contribuições de um grupo, relacionados com uma tarefa de um espaço de trabalho, por tipo de contribuição, exibidas como uma barra de gráficos.

Acima, foram citados apenas alguns exemplos. Existem várias outras possibilidades de pesquisa.

## Análise Qualitativa

Como foi citado anteriormente, a análise qualitativa apresenta três perspectivas. Os dados para análise dessas perspectivas vêm de três fontes: os dados da etapa da configuração de uma experiência, as contribuições dos aprendizes configuradas pelo sistema numa estrutura de árvore para cada tarefa do *workspace* e o conjunto de mensagens enviadas entre usuários no espaço de coordenação associado com cada atividade. O resultado da análise é representado através de um conjunto de características, pares atributos-valores, que são estabelecidos subjetivamente.

**Análise do Comportamento de Grupos** Na figura 3.7, é mostrado o conjunto de atributos usados para fazer a análise do comportamento de grupos. Nela é possível ver a existência dos atributos simples e atributos compostos. Valores para esses atributos são obtidos a partir dos dados de definição da tarefa e do processo executado, para atributos simples, e através da aplicação de inferências fuzzy, para os atributos compostos.

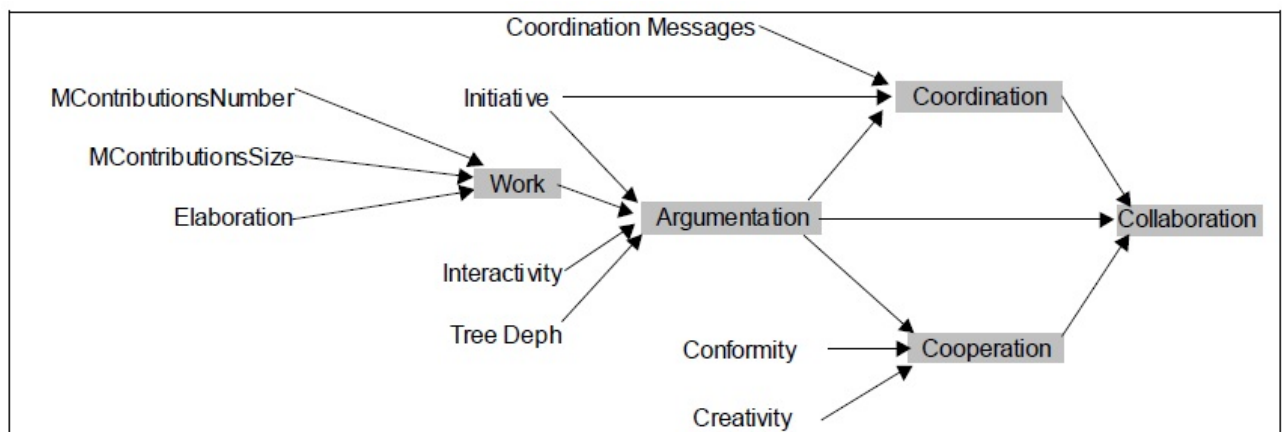


Figura 3.7: Atributos de análise de um grupo. Fonte: [10].

As variáveis *número de contribuições*, *tamanho das contribuições*, *profundidade da árvore de contribuições* e *interatividade* são definidas como:

- **Número médio de contribuições (MContributionsNumbers):** número de contribuições de um grupo/número de membros do grupo.
- **Tamanho médio das contribuições(MContributionsSize):** tamanho da contribuição (medido em caracteres).
- **Profundidade da árvore de contribuições (Tree Depth):** tamanho máximo das árvores relacionadas com esse espaço de trabalho.
- **Interatividade (Interactivity):** Porcentagem de contribuições respondidas ou ligadas com outras contribuições feitas por um estudante, outro além do contribuidor.

Durante a etapa de configuração, cada tipo de contribuição do modelo da estrutura de conversação foi valorado para os quatro atributos *Iniciativa*, *Elaboração*, *Criatividade* e *Conformidade*. Na fase de análise, a valoração de cada atributo é usado em uma equação



que definirá para esses mesmos atributos citados seus valores a nível de grupo, onde  $Va_i$  é o valor para um dos quatro atributos,  $N_j$  é a quantidade de contribuições para um determinado tipo,  $V_{ji}$  é o valor do atributo para um determinado tipo de contribuição.

$$Va_i = \sum_{j=1}^k (N_j * V_{ji})$$

Na análise qualitativa, os valores são inferidos utilizando métodos de inferência Fuzzy. A base da análise é composta pela definição dos conjuntos de atributos e as relações entre elas. Cada atributo foi modelado com um conjunto de etiquetas linguísticas. Cada uma dessas etiquetas foram marcadas com uma valor, por exemplo, “pouco”, “regular”, “muito”. Elas representam um alcance de valores possíveis, onde para cada atividade esses alcances podem variar. Dessa forma, para uma atividade pequena, um valor “x” pode ser valorado com a etiqueta “Alto”, entretanto, para atividades grandes, a etiqueta pode mudar para “Pouco”. A forma para julgar os alcances das etiquetas é através da comparação com outros grupos em outras atividades executadas do mesmo padrão.

**Análise Individual** A análise individual é realizada para cada estudante num grupo numa tarefa de aprendizagem, utilizando o mesmo método da análise do comportamento de grupos, entretanto, é utilizado um conjunto diferente de atributos e regras. Alguns atributos são calculados de dados da experiência para cada estudante no grupo levando em conta o valor médio para o grupo.

Os atributos para análise individual são: *número de contribuições adicionadas pelo estudante*, *o tamanho médio das contribuições*, e a *média da iniciativa, criatividade, elaboração e conformidade* de suas contribuições. Os atributos novos para análise individual são:

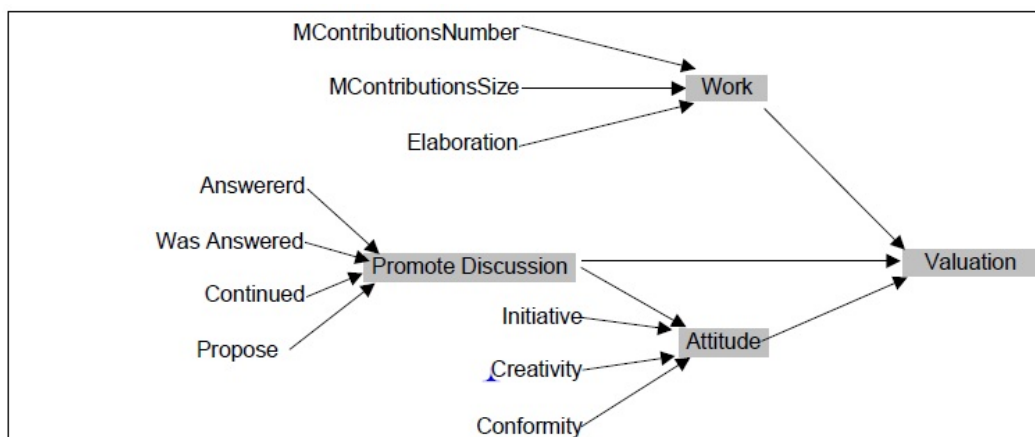


Figura 3.8: Atributos de análise de um indivíduo. Fonte: [10].

- O número de contribuições desse estudante que foram respondidas por outros;
- O número de contribuições de autoria de outros estudantes que foram respondidas por esses estudante;

- O número de contribuições de autoria desse estudante que foi continuada ou respondida por ele mesmo;
- O número de contribuições em estágio de “proposta” que foram elaboradas pelo estudante.

Esses atributos são combinados para inferir novos de modo a obter uma avaliação individual. A combinação dos atributos pode ser vista na imagem 3.8.

### 3.4.2 Modelo de Análise 3C

O modelo de análise 3C se baseia na visão de que a colaboração pode ser vista como a junção de três outros componentes, a cooperação, a comunicação e a coordenação. Ele é recorrente na literatura para classificar *groupwares*, como visto em [15].

A relação entre os 3Cs está ligada à interdependência entre eles. A cooperação gera artefatos de trabalho. As vezes esses artefatos por si só geram coordenação necessária para a sua continuidade. Entretanto, a complexidade do trabalho pode ser muito alta, sendo necessário utilizar alguma forma de comunicação explícita para que haja coordenação. Nem sempre a coordenação gera comunicação, e nem sempre a comunicação gera coordenação. O importante de perceber da relação entre esses três Cs é a criação da consciência do estado do trabalho. A seguir será feita a descrição de cada C separadamente.

#### Comunicação

A comunicação no paradigma colaborativo é declarada efetuada com sucesso quando o receptor entende o que o emissor enviou. Esse fato parece algo simples, mas permeia uma série de estruturas e elementos caracterizadores.

Ao iniciar o processo de comunicação, uma representação do que deve ser comunicado está na mente do emissor. É preciso transformar essa representação em algo real. Para tal, são utilizados elementos de expressão, que por exemplo, podem ser imagens, palavras. Entretanto, antes de se codificar a mensagem, é importante utilizar uma estrutura de linguagem para que haja coerência para o receptor.

Um segundo passo é enviar a informação através de um canal de percepção para o receptor. Através dos elementos de percepção, o receptor interpreta a mensagem, seguindo a estrutura de linguagem adequada.

Esse processo que foi descrito é a forma como se deve ser feita a comunicação. Além disso, pode-se ressaltar elementos que a caracterizam.

**Mídia:** Esse elemento se refere ao meio de comunicação da mensagem. Por exemplo, ela pode ser escrita, sonora, por meio de imagens, gestos, etc. Dependendo da mídia usada, pode-se restringir a capacidade de comunicar ao limitar as estruturas de linguagens que podem ser usadas.

**Modo de transmissão:** Define se a mensagem vai ser enviada em blocos ou continuamente. Isso implica nas possíveis ferramentas que podem ser utilizadas, por exemplo, *chats* de texto e fóruns para transmissões em bloco e videoconferências e *voip* para modo contínuo.

O importante com relação a parte humana é que a transmissão em blocos estimula o refinamento da mensagem antes de seus envios, pois permite reflexão e uma melhor preparação da mensagem antes do envio. No modo contínuo, estimula-se uma conversa mais fluída.

**Meta-informações e categoria:** A meta-informação apresenta dados que caracterizam a mensagem da comunicação. Permite ao usuário definir um contexto para uma mensagem, ao apresentar campos como data, título, identificadores de usuários, etc.

O elemento categoria serve para situar semanticamente o receptor. Por exemplo, permite ao receptor saber que tipo de resposta é esperada.

**Estrutura da conversa:** Define como as mensagens são estruturadas. Elas podem ser definidas como linear, hierárquica e em rede. O modo linear se caracteriza pelas mensagens montarem uma estrutura em lista, ou seja, uma mensagem está ligada somente a uma resposta.

O modo hierárquico (árvore) permite que a conversa sofra divergências no caminho, construindo assim uma relação de um nó pai permite mais de um filho. O modo em rede (grafo) permite que divergências nos caminhos possam se interconectar.

## Coordenação

A coordenação é o elemento que conecta os outros 2 Cs do modelo. Ela pode ser vista por trabalhar com três elementos, mais precisamente, pessoas, recursos e tarefas. A coordenação de pessoas está vinculada à comunicação e a de recursos está relacionada com a cooperação. Dessa maneira, ao falarmos exclusivamente da coordenação, é importante destacar a gestão de tarefas.

Ao falarmos de cooperação com enfoque nas tarefas, é preciso delimitar três etapas. A primeira é a pré-articulação, que envolve ações como identificação de objetivos, mapeamento de objetivos em tarefas, seleção de participantes e distribuição das tarefas.

A segunda é a gerência da atividade sendo executada. Ela é uma etapa dinâmica, pois exige interpretação dos estados das tarefas, análise e tomada de atitudes em curtos espaços de tempo.

A terceira é a pós-articulação que ocorre depois das execuções de todas as tarefas e envolve a documentação e análise das atividades.

Existem os níveis temporais e de objeto relacionados com a coordenação. Aquele descreve quais atividades devem ser executadas primeiro, determinando um fluxo de atividades. Este está relacionado com a coordenação de acessos sequenciais ou simultâneos por colaboradores à objetos que estão sendo trabalhados na cooperação.

## Cooperação

A cooperação é a operação conjunta num espaço de tempo e num espaço de trabalho compartilhado. Os colaboradores trabalham em conjunto editando, criando e organizando objetos. O espaço de trabalho fornece ferramentas para manipulação e gerência dos recursos permitindo ações como gravação de dados, recuperação de versões anteriores. O

versionamento está relacionado com a criação de uma memória das ações efetuadas durante a execução das tarefas, que será usado na etapa de pós-articulação. A coordenação e comunicação oferecem consciência do estado do trabalho sendo feito na cooperação.

### 3.5 Considerações Finais sobre o Capítulo

Nessa seção, foi feito um levantamento da importância dos itens abordados por este referencial teórico.

A primeira consideração é relacionada a colaboração apoiada por computador. As conferências aqui citadas serviram de base de pesquisa para esse projeto. A medida que foram feitas as pesquisas sobre sistemas colaborativos, foi possível nortear o rumo desse trabalho.

A segunda consideração é sobre o engageGrid e engageGrid+. É de extrema importância citar e explicar as características do projeto, de onde surgiu a necessidade dessa pesquisa, que no trabalho atual é a demonstração dos limites da API do Google Docs e de sua compatibilidade com a rede de compromissos.

A última consideração está relacionada com os modelos de análise. A literatura apresenta muitos sistemas que analisam trabalhos colaborativos. Porém, muitos dos trabalhos pesquisados não apresentavam uma estrutura bem definida de pesquisa. O DEGREE se mostrou bastante interessante, pois apresentou uma estrutura bem clara de análise que serviu como base para o trabalho atual. O modelo 3C permitiu analisar o ambiente de colaboração usado nessa pesquisa e assim delimitar as novas necessidades do trabalho.

## Capítulo 4

# Concepção e Prototipagem de Interface entre Serviços da Rede de Compromissos e Google Docs: Uma Primeira Versão

A proposta deste trabalho é o estudo de viabilidade entre a interface da rede de compromissos com a API do Google Docs. Ela segue a ideia da figura 4.1, onde os serviços da rede, que funcionam na nuvem, invocam os serviços do Google Docs quando necessários e posteriormente ocorre a recuperação de dados referentes aos artefatos produzidos na plataforma da Google.



Figura 4.1: Interação entre a rede de compromissos e a API do Google Docs.

Para demonstrar a viabilidade entre os serviços, foi proposto o desenvolvimento de um protótipo que possua o ciclo de vida de um compromisso e acesso aos serviços do Google

Docs. Com os requerimentos do ciclo de vida, sabe-se que são esperados os seguintes elementos da API do Google Docs:

- *Gerência de recursos*: Esse elemento está associado com a criação dos artefatos que serão produzidos na etapa de “Realização”.
- *Gerência de permissões*: Esse elemento está associado com a criação de papéis para os colaboradores durante a etapa de “Realização”.
- *Acesso aos dados do histórico de revisão*: Esse elemento está associado com a parte de análise automática da colaboração.

Tendo em vista os elementos citados anteriormente, as etapas de configuração, negociação e síntese não têm necessidades relacionadas com a API do Google Docs. Portanto, seus serviços são implementados no próprio protótipo, exceto a etapa de síntese, que não foi detalhadamente definida no projeto EngageGrid.

Um elemento importante a se destacar no protótipo é a necessidade de acesso ao histórico de revisão. A etapa de análise tem entre suas funções a análise implícita. Portanto, foi necessário as pesquisas relacionadas com modelos de análise da colaboração. Nesse trabalho foram usados dois modelos: modelo 3C [15] e o DEGREE [10]. O modelo de análise da colaboração 3C apresentou uma maneira de ver quais elementos presentes na API do Google Docs se fazem importantes no cálculo da colaboração e também quais elementos estão ausentes. O modelo de análise DEGREE apresenta uma estrutura organizada para se fazer o cálculo da colaboração (que é apresentado na segunda seção desse capítulo).

Esse capítulo está dividido em duas seções. A primeira seção apresenta a arquitetura do protótipo. A segunda apresenta o protótipo do ponto de vista de um usuário com apresentação das telas do sistema e do cálculo da colaboração.

## 4.1 Arquitetura

O sistema foi desenvolvido na linguagem JAVA. Essa escolha foi baseada em dois aspectos. O primeiro é a familiaridade com a linguagem e o segundo é a presença da biblioteca *SERVLET* que facilita a criação de aplicações *WEB*. Outro elemento usado nesse sistema foi o *web container* Apache Tomcat. Por questão de familiaridade, ele foi escolhido como ferramenta para servir como aplicação servidora.

A API do Google Docs apresentou até a data de realização dessa monografia três versões. A primeira e a segunda foram descartadas, pois não forneciam acesso ao histórico de revisão. Portanto, a versão 3.0 foi escolhida como API para acesso aos serviços do Google.

### 4.1.1 Organização Física

A figura 4.2 mostra como estão organizados os arquivos e pastas da aplicação. As cores apresentadas na figura possuem significados, onde o azul representa um recurso que seja uma pasta e o preto um arquivo. Por ser feito no Apache Tomcat, percebe-se sua organização com algumas pastas padrões como o “*WEB-INF*”, “*lib*” e “*classes*”. O importante a se destacar dessa organização são os seguintes elementos:

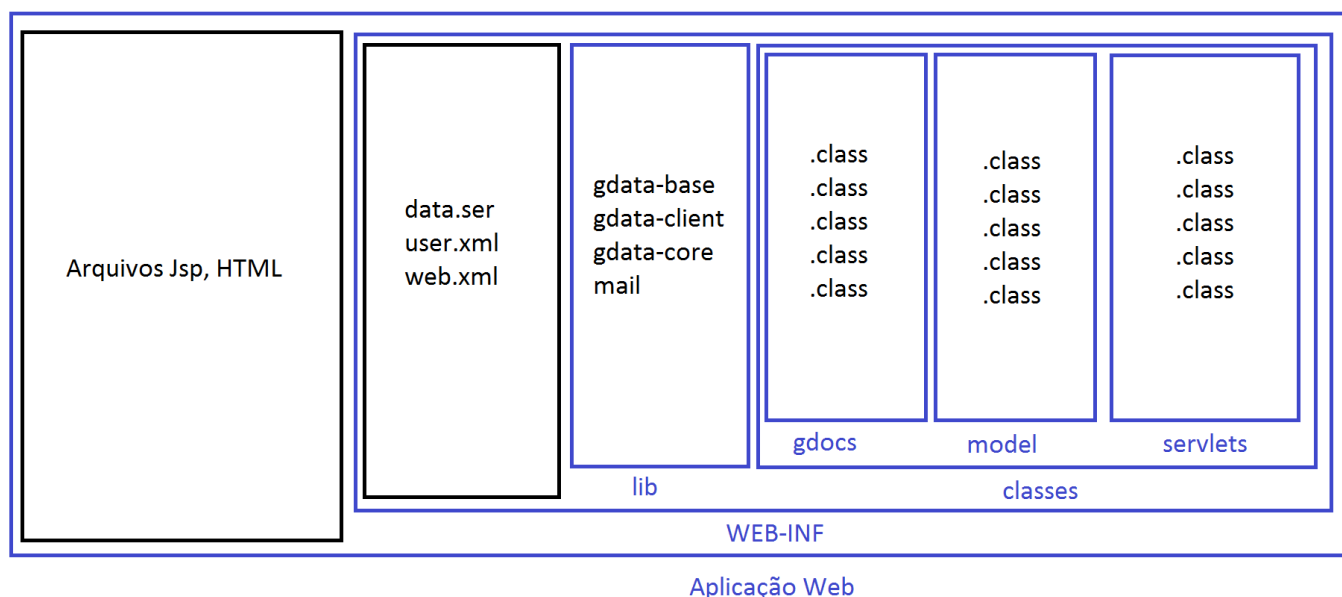


Figura 4.2: Organização dos recursos do protótipo.

- **Pasta da aplicação Web:** Essa pasta contém dois grupos de elementos importantes. O primeiro é o conjunto de arquivos JSP e HTML que serão responsáveis pelas telas que serão exibidas no navegador. O segundo elemento é a pasta WEB-INF, onde se encontram alguns arquivos de configuração, as bibliotecas e os arquivos “.class”.
- **Pasta “WEB-INF”:** Essa pasta contém a pasta “*lib*” com as bibliotecas usadas nessa aplicação, por exemplo, a biblioteca da API Google Docs 3.0. Contém também a pasta “*classes*”, onde estão todos os arquivos “.class” dessa aplicação. Além dessas pastas, há três arquivos de extrema importância para essa aplicação. O primeiro é o “data.ser”, que é um arquivo binário onde são armazenados todos os objetos da classe “webapp.src.model.CompromissoData.java”, que representa um compromisso criado na aplicação (Ela serve como classe modeladora de dados para armazenamento persistente). O segundo arquivo é o “user.xml”, que é o arquivo onde serão salvos todas as informações dos usuários cadastrados na aplicação. Esse arquivo armazena os seguintes dados: um *login*, uma senha, o email cadastrado no Google e um id. Para cadastrar qualquer usuário novo na aplicação, é necessária a edição manual desse arquivo citado anteriormente. O último arquivo é o “web.xml”, que é padrão usado no Tomcat para configuração dos mapeamentos entre os arquivos JSP, HTML ou *servlets* e as requisições enviadas para a aplicação.
- **Pasta “classes”:** Essa pasta é importante, pois é nela que contém os arquivos “.class” responsáveis pelos serviços apresentados na aplicação. Ela se divide em três pastas, cada qual com sua função, que são explicadas na próxima subseção.

### 4.1.2 Organização Conceitual

O protótipo apresenta uma organização conceitual do seus serviços distribuídos em três grupos, referentes as três pastas “servlets”, “model” e “gDocs” da organização física. Elas separam cada funcionalidade da aplicação que podem ser descritas como:

- **servlets:** Essa pasta contém todos as classes que implementam a interface “servlet”. Basicamente, cada tela (cada arquivo JSP) é representada por um arquivo servlet que trata da sua requisição. Existe uma classe chamada “*ServletBase*” que na verdade é a classe que implementa a Interface “*HttpServlet*”. Ela foi criada para padronizar as requisições GET feitas à aplicação e deixar somente as requisições POST customizáveis às classes que a estendem. Dessa maneira, todas as classes que tratam das requisições são subclasses da “*ServletBase*”. O pacote *Servlet* serve para tratar de todas as requisições que chegam na aplicação. Além de serem responsáveis pelas requisições, cada arquivo trata de uma etapa da rede de compromissos, tendo boa parte da lógica de negócio embutida nesses arquivos.
- **Model:** Essa pasta está relacionada com a montagem dos serviços básicos como a persistência e autenticação dos usuários. É utilizado o sistema de sessão padrão do Apache Tomcat. É nessa pasta que ficam as classes que modelam os tipos de dados usados na aplicação. Eles se referem aos dados: usuários, compromissos, sessão, unidade de revisão, colaboração, etc. O serviços dessa pasta são acessados através de invocações disparadas a partir da pasta *servlets*.
- **gDocs:** Essa pasta contém os arquivos responsáveis pela comunicação com a API e por realizar a operação de *parser* dos dados extraídos do “*changelog*”.

A figura 4.3 apresenta uma visão do fluxo de informação dentro da aplicação com destaque para os módulos citados. As setas hachuradas simbolizam o envio de informações, no caso, requisições. As setas não hachuradas simbolizam a resposta a uma informação. Os triângulos simbolizam arquivos presentes na aplicação.

## 4.2 Protótipo

O fluxo de telas é bem simples. Existe a tela de *login* (veja a figura 4.4) que é a tela de entrada para o sistema. Após a autenticação, tem-se acesso à tela de gerência dos compromissos. Ela é tela principal, pois é por ela que se acessa compromissos já criados e também é por ela que se cria novos compromissos. A partir do momento que é selecionado algum compromisso, o fluxo de telas segue a linearidade das etapas definidas no referencial teórico do EngageGrid [14], salvo quando é necessário fazer *login* na Google (veja a figura 4.5), que é requisitado através de uma tela específica.

As três telas que não estão relacionados com a parte conceitual citada anteriormente são a tela de *login* do sistema, a tela de gerência e a tela de permissão de acesso à API do Google Docs. A tela de *login* serve para permitir o acesso de usuários cadastrados no sistema. Os usuários são identificados através do arquivo “user.xml” presente no sistema. A tela de gerência é responsável por mostrar todos os compromissos presentes no sistema, como pode ser visto em 4.6. Ao lado de cada compromisso listado, aparece em que etapa



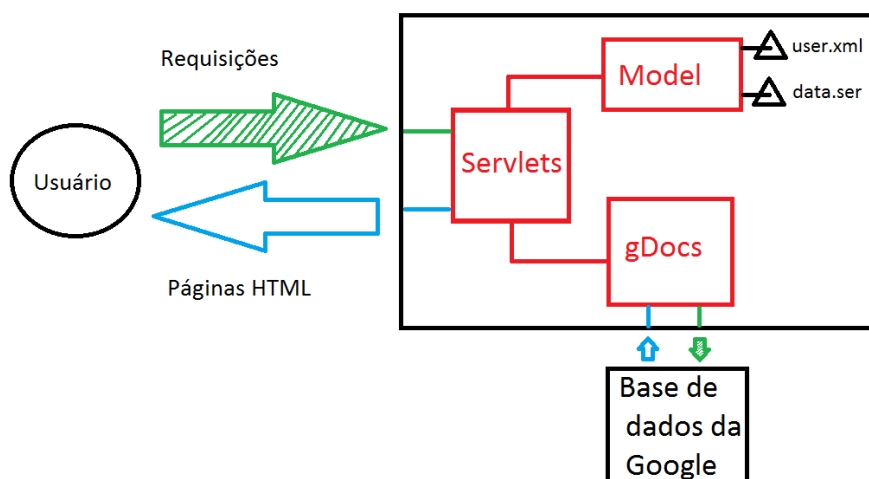


Figura 4.3: Arquitetura conceitual da aplicação que é representada pelos três módulos: “servlet”, “model” e “gDocs”.

do ciclo de vida do compromisso ele está. A tela de acesso à API serve para autorizar a aplicação a ter acesso aos dados do usuário na conta da Google.

A fase de configuração pretende-se que selecione alguns itens para criação de um compromisso. Portanto tela de configuração apresenta alguns campos básicos para cadastro de um compromisso. Esses campos são o nome, a data do fim, os colaboradores e medidas para checagem da colaboração, como pode ser visto em 4.7. Existem duas medidas para o cálculo da colaboração. Uma serve para limitar entre uma colaboração baixa/média e a outra serve para limitar entre uma colaboração média/alta.

A fase de negociação envolve a participação dos outros usuários através da confirmação e até a sugestão de outras ferramentas, temas, etc. Por ser uma etapa mais abrangente e envolver muitas variáveis que ainda não estão definidas, nesse trabalho ela foi resumida. Portanto, a tela dessa etapa mostra os dados cadastrados na etapa anterior e o único que pode concluir essa etapa é o usuário proponente do compromisso através do botão confirmar (esse botão aparece somente para o proponente, veja a figura 4.8). Após a confirmação do compromisso, caso o usuário não esteja logado na sua conta da Google dentro do sistema, serão solicitadas suas credenciais para tal. Após essa etapa, o documento será compartilhado com todos os usuários colaboradores e o compromisso muda para a etapa de realização.

A realização é a etapa onde serão executadas as tarefas pré-estabelecidas e confirmadas na negociação. Como as duas etapas iniciais foram simplificadas e o enfoque do protótipo é na ferramenta do Google Docs, a tela referente a essa etapa é bem simples. Nessa tela há um *link* que ao ser selecionado, leva o usuário à tela de edição da interface do usuário descrito na seção 3.3.1. Abaixo do link, há um campo que contém um *preview* do que já foi feito no documento. E abaixo do campo de preview, há um botão para fins práticos de mudança de etapa (Veja na figura 4.9).

Ao mudar de etapa, o sistema coleta todos os dados necessários para análise e posteriormente cancela o compartilhamento com os colaboradores, exceto o criador do documento. Isso acontece, pois é uma restrição da Google que exige que exista pelo menos um usuário

A login form titled 'Gestão dos compromissos'. It features a 'Login' heading, followed by input fields for 'Nome:' and 'Senha:'. Below these fields is a button labeled 'Enviar'.

Figura 4.4: Tela de *login* do protótipo.

A login form titled 'Gestão de compromissos' with the subtitle 'Google Docs API Login'. It includes the text 'Sua sessão com Google expirou ou ainda não foi conectada.' followed by input fields for 'Email:' (containing 'l.edwardmoreira@gmail.c') and 'Senha:'. A button labeled 'Enviar' is at the bottom.

Figura 4.5: Tela de *login* na API do Google Docs.

relacionado a um documento.

A última tela da aplicação é a de análise. É nela que são apresentados os resultados da colaboração efetuados na produção escrita. Somente os colaboradores que contribuíram para a aplicação aparecerão na tela de análise. Isso é uma limitação técnica que é explicada na próxima seção. Os dados que aparecem por colaborador são o número de caracteres inseridos, o número de caracteres removidos, a data e o horário do início e término da sessão para cada sessão feita pelo colaborador. É mostrado o desvio padrão para do número de inserções e a média de inserções. Veja na figura 4.10.

#### 4.2.1 Modelos de Análise e o Cálculo da Colaboração

O modelo 3C [15] foi usado como base para poder analisar a ferramenta de escrita colaborativa do Google de tal forma que cada uma dos 3 Cs fosse satisfeito parcialmente

Gestão dos compromissos

Lista de Compromissos

teste sessões! - Analise

Acessar Criar

Figura 4.6: Tela de gerência do protótipo.

Bem Vindo Eduardo Moreira

Gestão dos compromissos

Configuração

Nome do Compromisso: teste

Data de fim (dd/mm/aaaa): 1 / 1 / 2014

Valor para o desvio padrão aceito para colaboração média (em % da média): 50

Valor para o desvio padrão aceito para colaboração alta (em % da média): 25

Selecione os colaboradores:

Usuários	Colaboradores
Germana Nobrega	Edwarden Moreira
	Luiz Eduardo Moreira

Adicionar Remover

Enviar

Figura 4.7: Tela de configuração do protótipo.

ou até em sua integridade. Aqui nessa descrição, a satisfação parcial se daria pela presença de elementos da ferramenta que satisfaça as características descritas por cada elemento do 3C.

### Modelo 3C

O modelo 3C opera em cima de três variáveis: cooperação, comunicação e coordenação. Idealmente, um sistema que possuísse completamente esses três atributos, seria um sistema com uma grande capacidade de colaboração. Para analisar a ferramenta do Google Docs, foi realizada a separação de suas funcionalidades em relação aos três Cs.

- **Cooperação:** Esse elemento é representado pela área de edição e suas ferramentas de formatação. Esse elemento é o mais perceptível, pois o foco da ferramenta é a produção textual.

Gestão dos compromissos

Negociação

Nome do Compromisso: teste

Data de fim (dd/mm/aaaa): 1/1/2014

Colaboradores:  
Eduardo Moreira  
Edwarden Moreira  
Luiz Eduardo Moreira

Confirmar

Figura 4.8: Tela de negociação do protótipo.

Gestão de Compromissos

Tela Realização

Para acessar a página de edição do documento, clique no link abaixo.

[Clique aqui para acessar a tela de edição do documento](#)

Ir para Análise

Figura 4.9: Tela de realização do protótipo.

- Comunicação: Essa ferramenta possui o *chat* de conversa síncrono como forma direta de comunicação e o uso de emails para notificar o compartilhamento de documentos.
- Coordenação: A ferramenta apresenta o compartilhamento de documentos para gestão dos recursos, a definição de níveis de acesso aos documentos e comentários na área de edição do documento para coordenação de tarefas (que pode ser incluído como elemento de comunicação).

Ao analisar esses elementos presentes na ferramenta do Google Docs, percebe-se que ela apresenta bons elementos para colaboração. O elemento cooperação é bem representado por uma área de edição que permite a visualização da edição alheia. É carregado de informações que ajudam no fornecimento de consciência do que está acontecendo ou do que aconteceu (histórico de revisões).

Gestão dos compromissos

Análise

O usuário 1  
Inserções:0  
Deleções:0  
  
Sessões:

O usuário 2  
Inserções:1102  
Deleções:0  
  
Sessões:  
  
Id Sessão: 207b93de721a4fed  
Id Usuário: 06210642233680090865  
Data Inicial:Fri Jul 12 19:38:56 BRT 2013  
Data Final:Fri Jul 12 19:40:19 BRT 2013  
Duração (minutos): 1.3979167

O usuário 3  
Inserções:1060  
Deleções:0  
  
Sessões:  
  
Id Sessão: 327c734c03208f59  
Id Usuário: 17386727186718992294  
Data Inicial:Fri Jul 12 19:39:04 BRT 2013  
Data Final:Fri Jul 12 19:40:12 BRT 2013  
Duração (minutos): 1.1377833

Desvio Padrão: 509.87668  
  
Media de inserções: 720.6667  
  
O nível de colaboração é: Baixo

Figura 4.10: Tela de análise do protótipo.

O ferramenta de comunicação apresenta boa capacidade para transmitir informação, haja vista que é síncrona e se apresenta na mesma tela da edição do documento. A única dificuldade apresentada pela ferramenta em relação a comunicação é a falta de registro dessas mensagens. Ou seja, caso seja fechada a janela de edição, a informação que estava ali é perdida.

O elemento de coordenação de tarefas fica unicamente estabelecido através dos comentários. A coordenação de pessoas é facilmente executado pela comunicação utilizando o *chat*. A coordenação de recursos, pela proposta da ferramenta, é bem gerenciada pelo compartilhamento de recursos e a definição de níveis de acesso aos documentos.

A análise geral da ferramenta é que satisfaz bastante as necessidades de ambiente colaborativo. Óbvio que é importante ressaltar que há algumas situações em que a ferramenta não vai satisfazer completamente, porém ele é eficaz para proposta de escrita colaborativa. Vale ressaltar que a análise presente nessa monografia se refere somente

aos trabalhos escritos e que não foi definido qual vai ser o foco dos trabalhos produzidos na ferramenta. Portanto, o cálculo da colaboração foi definido de forma simples para satisfazer qualquer tipo de trabalho produzido.

## Modelo DEGREE

O modelo DEGREE [10] serviu como base para o cálculo da colaboração. Entretanto, é necessário fazer algumas observações. A primeira observação a se fazer é a diferença de foco dos projetos. O DEGREE foca na aprendizagem, enquanto, o EngageGrid tem foco no produto final. A segunda observação é em relação à ferramenta de comunicação. O DEGREE apresenta uma comunicação semi-estruturada com base em um grafo que a categoriza e define regras nas trocas de mensagens entre usuários.

O DEGREE busca analisar o grau de colaboração entre os participantes de uma atividade, pois essa colaboração denota maior integração ao realizar uma tarefa que envolve o aprendizado. Toda a estrutura para medir a colaboração vai usar medidas que estão envolvidas com o aprendizado. A busca pela análise da colaboração no Google Docs tem o enfoque no nível de comprometimento em realizar o trabalho em conjunto. É um foco diferente do apresentado no DEGREE, portanto as variáveis vão ter que sofrer mudanças para atender as necessidades. O campo de pesquisa da colaboração em escrita colaborativa é escasso para o molde aqui definido e não trouxe significativa contribuição para desenvolvimento de uma análise. As atividades feitas durante a escrita colaborativa não são triviais e demandam de uma análise mais detalhada que está fora do escopo dessa monografia.

Além da complexidade de analisar a colaboração com enfoque no trabalho, há algumas limitações com a ferramenta que impossibilitam a coleta de dados referentes a comunicação. Os dados do *chat* e dos comentários não estão disponíveis, ou seja, é possível se comunicar, porém não há como capturar esses dados para análise. Outra limitação que influencia na análise é o fato dos dados capturados do *changelog* serem somente referentes às atividades de edição dentro do documento. Isso interfere na capacidade de medir a interação dos colaboradores que tiveram maior atividade como leitores.

## Cálculo da Colaboração

Baseando-se na análise do modelo 3C e tendo como base o modelo de análise do trabalho DEGREE, esse protótipo apresenta o cálculo da colaboração baseado no número de inserções. A figura 4.11 mostra como foi a adaptação do modelo DEGREE à análise realizada no protótipo. As variáveis *creativity* (criatividade), *conformity* (conformidade), *initiative* (iniciativa), *elaboration* (elaboração), *coordination messages* (mensagens de coordenação), *interactivity* (interatividade), *tree deph* (profundidade da arvore de argumentação) e *MContributionsNumber* (número médio de contribuições) estão extremamente atrelados ao processo do DEGREE e não são aplicáveis as capacidades encontradas no Google Docs de acordo com modelo 3C. Dessa maneira, o cálculo se baseia somente na variável *MContributionSize* (média do tamanho das contribuições). No Google Docs, isso é representado pelo número de inserções.

Para chegar em um valor numérico, são realizados os seguintes cálculos:

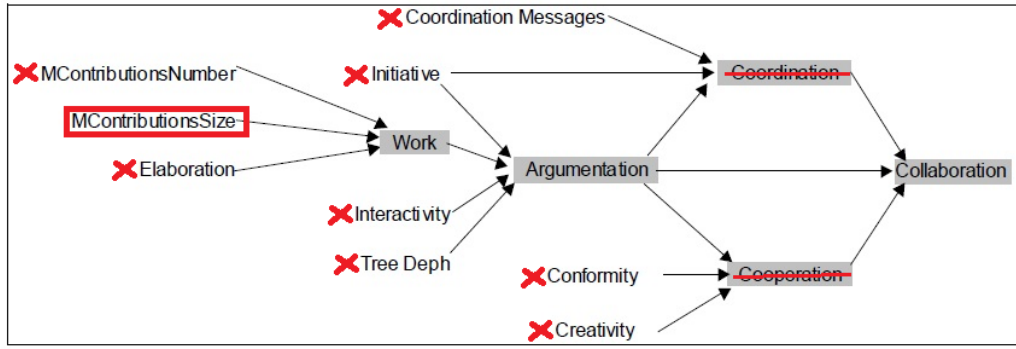


Figura 4.11: Reutilização do modelo DEGREE para este trabalho.

$$media = \sum_{i=1}^n \frac{numInsercao_i}{n}, n = numeroColaboradores$$

$$Variancia = \sum_{i=1}^n \frac{(numInsercao_i - media)^2}{n}, n = numeroColaboradores$$

$$DesvioPadrao = \sqrt{variancia}$$

$$graudesimetria = \frac{desvioPadrao}{media}$$

A última operação obtém a porcentagem do desvio padrão em relação à média de inserções. Sabe-se que o desvio padrão fornece o valor de quanto os valores usados para calcular a média desviam dela mesma. Portanto, quanto mais próximo de zero for o valor do desvio padrão, mais próximos os valores das inserções estarão em relação umas das outras. Assim, o cálculo é baseado no nível de simetria das inserções dos colaboradores.

Na etapa de configuração, é dado ao proponente a possibilidade de definir quanto do valor do grau de simetria é permitido para se ter uma colaboração alta, média e baixa. Portanto, se o grau de simetria for menor que o valor estabelecido para a variável de colaboração alta, tem-se a colaboração alta. E dessa maneira, são estabelecidos os níveis de colaboração. Para entender melhor, veja abaixo:

$$Graudesimetria < valorColaboracaoAlta$$

, então o nível de colaboração foi alto.

$$ValorColaboracaoMedio > Graudesimetria \geq valorColaboracaoAlta$$

, então o nível de colaboração foi médio.

$$ValorColaboracaoMedio \leq Graudesimetria$$

, então o nível de colaboração foi baixo.

# Capítulo 5

## Conclusão

### 5.1 Conclusão

Essa monografia de graduação propôs a construção de um protótipo capaz de acessar os dados do histórico de revisão e efetuar a análise da colaboração de um documento produzido no Google Docs.

O protótipo desenvolvido mostrou como os elementos importantes da API do Google se adaptaram às etapas do ciclo de vida de um compromisso. Através dos modelos DEGREE e 3C foi possível analisar os elementos colaborativos presentes no ambiente de colaboração e na API. A partir dessa análise, foi possível delimitar quais variáveis do modelo DEGREE poderiam ser utilizadas, além de verificar quais elementos do ambiente colaborativo se fazem importantes para compor o modelo.

O cálculo da colaboração foi restringido pela limitação na captura dos dados referentes à comunicação. O Google Docs apresenta ferramentas para comunicação, entretanto, não é possível ter acesso aos dados inseridos nelas. Diante da limitação, o cálculo é medido através da simetria de inserções de caracteres por cada usuário.

A grande dificuldade desse projeto foi a descoberta de uma maneira de capturar os dados do histórico de revisão. A documentação apresentada pelo Google não tinha referência a esse tipo de captura. Outro fator que atrapalhou a pesquisa foi o fato da página que continha a documentação referente a essa API conter somente exemplos das respostas HTTP e exemplos de códigos na plataforma .NET. Além disso, os programas exemplo fornecidos na linguagem JAVA apresentaram problemas durante sua execução, o que dificultou o estudo do comportamento da API.

A busca por um modelo de análise percorreu todos os simpósios descritos no capítulo 3. O esperado era achar algum modelo de análise na área de CSCW. Porém, a maioria dos trabalhos encontrados não analisavam o trabalho ou o processo executado pelas ferramentas descritas. A área de CSCL apresenta um maior número de trabalhos contendo análises, porém, nenhum trabalho que se adaptasse ao Google Docs e em sua maioria os modelos estavam muito atrelados às ferramentas desenvolvidas.

Apesar da limitação do cálculo e das dificuldades encontradas, o objetivo de estabelecer uma interface entre a rede compromissos e o objetivo de prover a análise da colaboração foram alcançados.



## 5.2 Trabalhos Futuros

A seguir são listadas algumas propostas de trabalhos futuros.

**Análise dos processos de escrita e as suas variáveis que afetam o trabalho colaborativo.** Os trabalhos apresentados no capítulo 2 dessa monografia apresentam uma nova abordagem na escrita colaborativa. Através dos processos ali desenvolvidos, fazer um estudo para desenvolver variáveis que possam ser usadas no mensuramento da colaboração.

**Revisão das possibilidades de colaboração com a API Google DRIVE** A versão da API do Google Docs apresentada nesse trabalho foi depreciada no meio do projeto. Um novo trabalho pode ser feito para estudar as capacidades e limites dessa nova API.

# Referências

- [1] Criwg 2013. "<http://criwg2013.vuw.ac.nz/>". 16
- [2] Csc1 2013. "<http://www.isls.org/csc12013/>". 16, 33
- [3] Cscw 2012. "<http://www.cscw2012.org/>". 15
- [4] ecscw 2013. "<http://ecscw2013.cs.ucy.ac.cy/>". 16
- [5] Google docs. "[https://support.google.com/drive/answer/143206?ref\\_topic=21008&rd=1](https://support.google.com/drive/answer/143206?ref_topic=21008&rd=1)". 1, 14, 19
- [6] Google documents list api version 3.0. "<https://developers.google.com/google-apps/documents-list/>". Acessado em 12/05/12. 1, 23
- [7] Office web apps. "<http://office.microsoft.com/pt-br/web-apps/>". 19
- [8] Sbsc 2013. "[http://shine.icomp.ufam.edu.br/sbsc\\_ihc2013/sbsc/](http://shine.icomp.ufam.edu.br/sbsc_ihc2013/sbsc/)". 17
- [9] Tml. "<http://tml-java.sourceforge.net/>". 7
- [10] B. Barros and M. F. Verdejo. Analysing student interaction processes in order to improve collaboration: the degree approach. *Int. J. AIED*, pages 221–241, 2000. vii, 32, 33, 35, 37, 38, 43, 51
- [11] H. Bonanno and J. Jones. *Measuring the Academic Skills of University Students - the MASUS procedure, a diagnostic assessment*. University of Sydney, 1997. 4
- [12] G. M. da Nóbrega. Projeto cloudnet: Apoio à configuração de compromissos profissionais para realização em rede e em nuvem. *Plataforma Carlos Chagas CNPq*, 2011. 1, 18, 19
- [13] G. M. da Nóbrega. Projeto engagegrid+: Plano de trabalho para bolsista dti. *Plataforma Carlos Chagas CNPq*, 2012. 19
- [14] G. M. da Nóbrega, F. Lima, and F. W. Cruz. Rumo a uma rede de compromissos em desktop compartilhado: aspectos conceituais e tecnológicos. *SBSC 2010*, pages 1–4, 2010. vii, 17, 19, 45
- [15] H. Fuks, A. B. Raposo, M. A. Gerosa, and C. J. P. Lucena. Applying the 3c model to groupware development. *International Journal of Cooperative Information Systems*, pages 299–328, 2005. 39, 43, 47

- [16] A. Goldberg, M. Russell, and A. Cook. *The Effect of Computers on Student Writing: A Meta-analysis of Studies from 1992 to 2002*. Boston College, 2002. 7
- [17] Y. Gong. Generic text summarization using relevance measure and latent semantic analysis. In *in Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001. 4
- [18] J. Grudin. Computer-supported cooperative work: History and focus. pages 19–26, 1994. 15
- [19] P. B. Lowry, A. Curtis, and M. R. Lowry. Building a taxonomy and nomenclature of collaborative writing to improve interdisciplinary research and practice. *Journal of Business Communication*, pages 66–99, 2009. 7, 9
- [20] S. Osinski, D. Weiss, and J. Stefanowski. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, pages 48–54, 2005. 4
- [21] V. Southavilay, K. Yacef, and R.A. Calvo. Writeproc: A framework for exploring collaborative writing processes. 2009. vii, 3, 6, 7, 8, 9
- [22] V. Southavilay, K. Yacef, and R.A. Calvo. Process mining to support students' collaborative writing. *Educational Data Mining conference proceedings*, 2010. vii, viii, 8, 10, 12, 13
- [23] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The prom framework: a new era in process mining tool support. In *Proceedings of the 26th international conference on Applications and Theory of Petri Nets*, ICATPN'05, pages 444–454, Berlin, Heidelberg, 2005. 7
- [24] J. Villalón, P. Kearney, R. A. Calvo, and P. Reimann. Glosser: Enhanced feedback for student writing tasks. In *Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies*, ICALT '08, pages 454–458, Washington, DC, USA, 2008. IEEE Computer Society. vii, 3, 4, 5, 6, 14